

DV11

DV11 ROM TST PRT 1
CZDVCD0

AH-8737D-MC

SEP 1979

COPYRIGHT · 75-79

digital

FICHE 1 OF 1

MADE IN USA

801

EDF1CZD*IBSEQ
PCF10 411 SEQ 0001

00010000

780330

PDP10 411
IDENTIFICATION

HDR1CZDVDCSEQ

00010000

780330

PRODUCT CODE: AC-8740C-MC
PRODUCT NAME: CZDVDCO DV11 ROM TS* PR12
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)
MSR 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 BK OF MEMORY EXCEPT WHERE ABL AND FOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1736 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER DV11 TRIAL PROGRAM HAS BEEN EXECUTED; OR AFTER THE 'AUTO SIZING' HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA SUCH AS DISK, MAGTAPE, DECTAPE, OR CASSETTE, FOLLOW INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY * SIZE

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

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER.
(ALSO PLACE 'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW BE LOADING INTO CPU)

4. STARTING PROCEEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR 'AUTO SIZING' OR LEAVE
LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP BY DV11 TRIAL PROGRAM OR A PREVIOUSLY RUN DV11 DIAGNOSTIC THAT USED THE 'AUTO SIZING'. (SECTION 7.2 AND 8.4, 8.5 MAY BE HELPFUL)
- D. DEPRESS 'START KEY' AND RELEASE THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

```

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

```

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC

4.1 CONTROL SWITCH SETTINGS

NOTE: IF THERE IS NO REAL SWR (177570); SWR MAY BE MODIFIED AT LOC:176 OR BY HITTING CONTROL "G" (↑G) ON CONSOLE TERMINAL.

```

SW 15 SET: HALT ON ERROR
SW 14 SET: LOOP ON CURRENT TEST
SW 13 SET: INHIBIT ERROR PRINT OUT
SW 12 SET: INHIBIT **ALL** TYPE OUT/BELL ON ERROR.
SW 11 SET: INHIBIT ITERATIONS. (QUICK PASS)
SW 10 SET: ESCAPE TO NEXT TEST
SW 09 SET: LOOP WITH CURRENT DATA
SW 08 SET: CATCH ERROR AND LOOP ON IT
SW 07 SET: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.
SW 06 SET: RESERVED
SW 05 SET: RESERVED
SW 04 SET: RESERVED
SW 03 SET: RESERVED
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: RESELECT DV11'S DESIRED ACTIVE.

```

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DV11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DV11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS FOUR DV11S; BITS 00,01,02,03 WILL BE SET IN LOC 'DVACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW00) ALTERS THAT LOCATION; THEREFORE IF FOUR DV11S ARE IN THE SYSTEM ***DO NOT*** SET 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 HASS 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 PRIORITIES

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 'SCOPI') 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 ENABLED; AND THERE IS A *HARD* ERROR (CONSTANT); SW08 IS BEST. (SW14=1, SW10=0, SW09=0, SW08=1). FOR INTERMITTENT ERRORS; SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NOT ERROR. (SW14=1, SW10=0, SW09=0, SW08=1,0)
2. SW 14
3. SW 11

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200 THERE ARE NO OTHER STARTING ADDRESSES FOR THE DV11 DIAGNOSTICS PREVIOUSLY MENTIONED EXCEPT FOR DZDVE WHICH IS: 000200 FOR THE MODEM CONTROL AND CABLE TESTS AND 000210 FOR THE MANUAL PARAMETER INPUT PROGRAM.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY AFTER *ALL* AVAILABLE DV11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5. OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED.

AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC

5.2 PROGRAM AND OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15=1) WHEN EVER AN ERROR OCCURS.
2. CLEAR SW 15.
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST) TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT; LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE 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.

5. 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.

5.2 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.

2.4 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.

D/SCR (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.

B.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.

2.5 *** METHOD OF AUTO SIZING ***

2.5.1 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 MXM 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.

2.5.2 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.

2.5.3 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.
2. SET BIT15 OF STATUS MAP OF ANY (APPROIATE) LINE CARDS MISSING TWO SYNC.
3. SET BIT12 IF YOU HAVE A 4 LINE GROUP SET FOR 1 SYNC.
4. EIGHT BITS PER CHAR.
5. ADJUST BITS 9 AND BIT 8 IN STATUS MAP FOR YOUR CORRECT CONFIG.
6. SYNCHRONOUS LINE CARDS INSTALLED
7. SET BIT11 OF STATUS MAP FOR ASYNC LINE CARD AND ZERO SYNC CHARS.
8. SYNC "A"=226 AND SYNC "B"=062

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

MO1

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 2
CZDVDC.P11 02-FEB-78 13:55

SEG 0012

;*AC-B740C-MC/(<377>)/"CZDVDCO DV11" ROM TST PRT2
;*COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
;-----

; STARTING PROCEDURE
; LOAD PROGRAM
; LOAD ADDRESS 000200
; PRESS START
; PROGRAM WILL TYPE "AC-B740C-MC/(<377>)/"CZDVDCO 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
;-----

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

SW15=100000
SW14=40000
SW13=20000
SW12=10000
SW11=4000
SW10=2000
SW09=1000
SW08=400
SW07=200
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

;=1, HALT ON ERROR
;=1, LOOP ON CURRENT TEST
;=1, INHIBIT ERROR TYPEOUT
;=1, DELETE TYPEOUT/BELL ON ERROR.
;=1, INHIBIT ITERATIONS
;=1, ESCAPE TO NEXT TEST ON ERROR
;=1, LOOP WITH CURRENT DATA
;=1, LOOP ON ERROR
;=1, DO "AUTO SIZING" ON INITIAL START UP.

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

000000
000001
000002
000003
000004
000005
000006
000007
177776
001200
100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001
010000
020000
030000
040000
050000
060000
070000

; 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
;-----

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

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
93
94  : STANDARD INTERRUPT VECTORS
95  :-----
96
97  000024      .=24      .PFAIL      : POWER FAIL HANDLER
98  000026      004402      340      : SERVICE AT LEVEL 7
99  000030      004002      .HLT      : ERROR HANDLER
100 000032      000340      340      : SERVICE AT LEVEL 7
101 000034      003750      .TRPSRV   : GENERAL HANDLER DISPATCH SERVICE
102 000036      000340      340      : SERVICE AT LEVEL 7
103
104 000040      000001      .=40      .BLKW 1      : SAVE FOR ACT-11 OR DDP2
105 000042      000001      .BLKW 1      : RETURN ADDRESS IF UNDER ACT-11 OR DDP2
106 000044      000001      .BLKW 1      : SAVE FOR ACT-11 OR DDP2
107 000046      002560      LOGICAL      : FOR USE WITH ACT-11 OR DDP2
108
109
110 000174      000174      .=174      LIGHT: 0
111 000176      000176      .=176      SSWR: 0
112
113
114 000200      000200      .=200      JMP      .START      : GO TO START OF PROGRAM
115 000200      000137      001742
116
117
118 001000      001000      .=1000      MTITLE: .ASCIZ  <377><12>/AC-B740C-MC/<377>/"CZDVDCO DV11" ROM TST PRT2.<377>
119 001000      005377      041501  034055
120
121 001200      001200      .=1200      LIGHTS:
122 001200      177570      177570      SWR:
123 001202      177570      177570      : INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
124  :-----
125
126
127 001204      177560      TKCSR: 177560      : TELETYPE KEYBOARD CONTROL REGISTER
128 001206      177562      TKDBR: 177562      : TELETYPE KEYBOARD DATA BUFFER
129 001210      177564      TPCSR: 177564      : TELEPRINTER CONTROL REGISTER
130 001212      177566      TPDBR: 177566      : TELEPRINTER DATA BUFFER
131
132  : PROGRAM CONTROL PARAMETERS
133  :-----
134
135 001214      000000      RETURN: 0      : SCOPE ADDRESS FOR LOOP ON TEST
136 001216      000000      NEXT: 0      : ADDRESS OF NEXT TEST TO BE EXECUTED
137 001220      000000      LOCK: 0      : ADDRESS FOR LOCK ON CURRENT DATA

```

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 5
 CZDVDC.P11 02-FEB-78 13:55

SEG 0015

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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	SYNEX: 0	
150	001242	000000	CLKX: 0	
151	001244	000000	MASKX: 0	
152	001246	000000	TEMP1: 0	; TEMPORARY STORAGE
153	001250	000000	TEMP2: 0	; TEMPORARY STORAGE
154	001252	000000	TEMP3: 0	; TEMPORARY STORAGE
155	001254	000000	TEMP4: 0	; TEMPORARY STORAGE
156	001256	000000	TEMP5: 0	; TEMPORARY STORAGE
157	001260	000000	SAVR0: 0	; R0 STORAGE
158	001262	000000	SAVR1: 0	; R1 STORAGE
159	001264	000000	SAVR2: 0	; R2 STORAGE
160	001266	000000	SAVR3: 0	; R3 STORAGE
161	001270	000000	SAVR4: 0	; R4 STORAGE
162	001272	000000	SAVR5: 0	; R5 STORAGE
163	001274	000000	SAVSP: 0	; STACK POINTER STORAGE
164	001276	000000	SAVPC: 0	; PROGRAM COUNTER STORAGE
165	001300	000001	DVACTV: .BLKB 1	; DV11'S SELECTED ACTIVE.
166	001301	000001	DVNUM: .BLKB 1	; OCTAL NUMBER OF DV11'S.
167	001302	000001	SAVACT: .BLKB 1	; ORIGINAL ACTV. DEVICES.
168	001303	000001	SAVNUM: .BLKB 1	; WORKABLE NUMBER.
169	001304	000001	RUN: .BLKB 1	; POINTER ONE PAST RUNNING DEVICE.
170		001306	.EVEN	
171	001306	001500	CREAM: DV.MAP	; TABLE POINTER.

CZDVDC0 MACY11 30A(1052) 10-FEB-78 15:44 PAGE 6
 CZDVDC.P11 02-FEB-78 13:55 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

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

```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

;DV11 VECTOR AND REGISTER INDIRECT POINTERS

224					
225					
226	001352	000000	DVRVEC: 0	; POINTER TO DV11 RECEIVER INTERRUPT VECTOR	
227	001354	000000	DVRLVL: 0	; POINTER TO DV11 RECEIVER INTERRUPT SERVICE PS	
228	001356	000000	DVTVEC: 0	; POINTER TO DV11 TRANSMITTER INTERRUPT VECTOR	
229	001360	000000	DVTLVL: 0	; POINTER TO DV11 TRANSMITTER INTERRUPT SERVICE PS	
230	001362	000000	DVSCR: 0	; POINTER TO DV11 SYSTEM CONTROL REGISTER	
231	001364	000000	DVSCRH: 0	; POINTER TO DV11 SYSTEM CONTROL REGISTER HIGH BYTE.	
232	001366	000000	DVRIC: 0	; POINTER TO DV11 NEXT RECEIVED CHARACTER REGISTER	
233	001370	000000	DVLCR: 0	; POINTER TO DV11 LINE PRAMETER REGISTER	
234	001372	000000	DVSRS: 0	; POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER	
235	001374	000000	DVSRSH: 0	; POINTER TO DV11 SECONDARY REGISTER SELECT HIGH BYTE.	
236	001376	000000	DVSRA: 0	; INTER TO DV11 SECONDARY REGISTER ACCESS REGISTER	
237	001400	000000	DVSFR: 0	;)INTER 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					

;DV11 CONTROL INDICATORS FOR CURRENT DV11 UNDER TEST

242					
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	000	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 8.	; NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03	
251	001413	010	CLK.B: .BYTE 8.	; NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07	
252	001414	010	CLK.C: .BYTE 8.	; NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11	
253	001415	010	CLK.D: .BYTE 8.	; 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					

; SUMMARY

265					
266					
267					
268					
269					
270					
271					
272					
273					
274					
275					

:	MASK.X	040	5 BITS PER CHAR.
:		100	6 BITS PER CHAR.
:		200	7 BITS PER CHAR.
:		000	8 BITS PER CHAR.
:			
:	CLK.X	005	5 BITS PER CHAR.
:		006	6 BITS PER CHAR.
:		007	7 BITS PER CHAR.
:		010	8 BITS PER CHAR.

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

276                                     ;DV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
277                                     ;-----
278
279                                     . =1500
280 001500 000001 DV.MAP: .BLKW 1
281 001500 000001 DVCRO0: .BLKW 1
282 001502 000001 DVTR00: .BLKW 1
283 001504 000001 DV00.A: .BLKW 1
284 001506 000001 SYNA00: .BLKW 1
285 001510 000001 DV00.B: .BLKW 1
286 001512 000001 SYN000: .BLKW 1
287 001514 000001 DV00.C: .BLKW 1
288 001516 000001 SYNC00: .BLKW 1
289 001520 000001 DV00.D: .BLKW 1
290 001522 000001 SYND00: .BLKW 1
291
292 001524 000001 DVCRO1: .BLKW 1
293 001526 000001 DVTR01: .BLKW 1
294 001530 000001 DV01.A: .BLKW 1
295 001532 000001 SYNA01: .BLKW 1
296 001534 000001 DV01.B: .BLKW 1
297 001536 000001 SYN001: .BLKW 1
298 001540 000001 DV01.C: .BLKW 1
299 001542 000001 SYNC01: .BLKW 1
300 001544 000001 DV01.D: .BLKW 1
301 001546 000001 SYND01: .BLKW 1
302
303 001550 000001 DVCRO2: .BLKW 1
304 001552 000001 DVTR02: .BLKW 1
305 001554 000001 DV02.A: .BLKW 1
306 001556 000001 SYNA02: .BLKW 1
307 001560 000001 DV02.B: .BLKW 1
308 001562 000001 SYN002: .BLKW 1
309 001564 000001 DV02.C: .BLKW 1
310 001566 000001 SYNC02: .BLKW 1
311 001570 000001 DV02.D: .BLKW 1
312 001572 000001 SYND02: .BLKW 1
313
314 001574 000001 DVCRO3: .BLKW 1
315 001576 000001 DVTR03: .BLKW 1
316 001600 000001 DV03.A: .BLKW 1
317 001602 000001 SYNA03: .BLKW 1
318 001604 000001 DV03.B: .BLKW 1
319 001606 000001 SYN003: .BLKW 1
320 001610 000001 DV03.C: .BLKW 1
321 001612 000001 SYNC03: .BLKW 1
322 001614 000001 DV03.D: .BLKW 1
323 001616 000001 SYND03: .BLKW 1
324
325 001620 000001 DVCRO4: .BLKW 1
326 001622 000001 DVTR04: .BLKW 1
327 001624 000001 DV04.A: .BLKW 1
328 001626 000001 SYNA04: .BLKW 1
329 001630 000001 DV04.B: .BLKW 1
330 001632 000001 SYN004: .BLKW 1
331 001634 000001 DV04.C: .BLKW 1

```

CONTROL STATUS REGISTER FOR DV11 NUMBER 00
VECTOR "A" FOR DV11 NUMBER 00
PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
SYNC TWO
PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
SYNC TWO
PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
SYNC TWO
PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
SYNC TWO

CONTROL STATUS REGISTER FOR DV11 NUMBER 01
VECTOR "A" FOR DV11 NUMBER 01
PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
SYNC TWO
PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
SYNC TWO
PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
SYNC TWO
PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
SYNC TWO

CONTROL STATUS REGISTER FOR DV11 NUMBER 02
VECTOR "A" FOR DV11 NUMBER 02
PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
SYNC TWO
PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
SYNC TWO
PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
SYNC TWO
PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
SYNC TWO

CONTROL STATUS REGISTER FOR DV11 NUMBER 03
VECTOR "A" FOR DV11 NUMBER 03
PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
SYNC TWO
PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
SYNC TWO
PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
SYNC TWO
PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 03
SYNC TWO

CONTROL STATUS REGISTER FOR DV11 NUMBER 04
VECTOR "A" FOR DV11 NUMBER 04
PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
SYNC TWO
PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
SYNC TWO
PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

332	001636	000001	SYNCO4: .BLKW 1	:SYNC TWO
333	001640	000001	DVO4.D: .BLKW 1	:PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
334	001642	000001	SYND04: .BLKW 1	:SYNC TWO
335				
336	001644	000001	DVCR05: .BLKW 1	:CONTROL STATUS REGISTER FOR DV11 NUMBER 05
337	001646	000001	DVTR05: .BLKW 1	:VECTOR "A" FOR DV11 NUMBER 05
338	001650	000001	DVO5.A: .BLKW 1	:PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
339	001652	000001	SYNA05: .BLKW 1	:SYNC TWO
340	001654	000001	DVO5.B: .BLKW 1	:PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
341	001656	000001	SYNB05: .BLKW 1	:SYNC TWO
342	001660	000001	DVO5.C: .BLKW 1	:PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
343	001662	000001	SYNCO5: .BLKW 1	:SYNC TWO
344	001664	000001	DVO5.D: .BLKW 1	:PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
345	001666	000001	SYND05: .BLKW 1	:SYNC TWO
346				
347	001670	000001	DVCR06: .BLKW 1	:CONTROL STATUS REGISTER FOR DV11 NUMBER 06
348	001672	000001	DVTR06: .BLKW 1	:VECTOR "A" FOR DV11 NUMBER 06
349	001674	000001	DVO6.A: .BLKW 1	:PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
350	001676	000001	SYNA06: .BLKW 1	:SYNC TWO
351	001700	000001	DVO6.B: .BLKW 1	:PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
352	001702	000001	SYNB06: .BLKW 1	:SYNC TWO
353	001704	000001	DVO6.C: .BLKW 1	:PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
354	001706	000001	SYNCO6: .BLKW 1	:SYNC TWO
355	001710	000001	DVO6.D: .BLKW 1	:PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
356	001712	000001	SYND06: .BLKW 1	:SYNC TWO
357				
358	001714	000001	DVCR07: .BLKW 1	:CONTROL STATUS REGISTER FOR DV11 NUMBER 07
359	001716	000001	DVTR07: .BLKW 1	:VECTOR "A" FOR DV11 NUMBER 07
360	001720	000001	DVO7.A: .BLKW 1	:PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
361	001722	000001	SYNA07: .BLKW 1	:SYNC TWO
362	001724	000001	DVO7.B: .BLKW 1	:PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
363	001726	000001	SYNB07: .BLKW 1	:SYNC TWO
364	001730	000001	DVO7.C: .BLKW 1	:PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
365	001732	000001	SYNCO7: .BLKW 1	:SYNC TWO
366	001734	000001	DVO7.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
372
373
374
375
376
377
378 001742 012737 000340 177776 .START: MOV #340,PS
379 001750 012706 001200 MOV #STACK,SP
380 001754 012737 004402 000024 MOV #.PFAIL,2#24
381 001762 113737 001301 001303 MOV DVNUM,SAVNUM
382 001770 005037 001230 CLR PASCNT
383 001774 105037 001311 CLRB ERRFLG
384 002000 105037 001313 CLRB QV.FLG
385 002004 012737 001500 001306 MOV #DV.MAP,CREAM
386 002012 112737 000001 001304 MOVB #1,RUN
387 002020 005037 001232 CLR ERRCNT
388 002024 005037 001234 CLR LSTERR
389 002030 012737 000001 001226 MOV #1,TSINO
390 002036 012737 001742 001214 MOV #.START,RETURN
391
392 002044 105737 001310 TSTB INIFLG
393 002050 001063 BNE 1$
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$,4
398 002074 005777 177102 TST 2SWR
399 002100 000240 NOP
400 002102 000407 BR 81$
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 ,MTITLE
407 002136 105137 001310 COMB INIFLG
408 002142 105777 177034 TSTB 2SWR
409 002146 100402 BMI 16$
410 002150 004737 006624 JSR PC,CSRMAP
411 002154 104402 005461 16$: TYPE ,XHEAD
412 002160 012737 001500 001246 MOV #DV.MAP,TEMP1
413 002166 017737 177054 001250 5$: MOV 2TEMP1,TEMP2
414 002174 022737 177777 001250 CMP #177777,TEMP2
415 002202 001406 BEQ 1$
416 002204 104410 CONVRT
417 002206 005506 XSTATQ
418 002210 062737 000002 001246 ADD #2,TEMP1
419 002216 000763 BR 5$
420 002220 005737 000042 1$: TST 2#42
421 002224 001030 BNE 3$
422 002226 032777 000001 176746 BIT #SW00,2SWR
423 002234 001424 BEQ 3$
424 002236 104402 005402 TYPE ,MNEW
425 002242 005000 CLR RD

```

```

:PROGRAM INITIALIZATION
:LOCK OUT INTERRUPTS
:SET UP PROCESSOR STACK
:SET UP POWER FAIL VECTOR
:CLEAR PROGRAM CONTROL FLAGS AND COUNTS
:TYPE TITLE MESSAGE

```

```

:LOCK OUT INTERRUPTS
:SET UP STACK
:SET UP POWER FAIL VECTOR
:SAVE NUMBER OF DEVICES IN SYSTEM.
:CLEAR PASS COUNT
:CLEAR ERROR FLAG
:ZERO QUICK VERIFY FLAG
:GET MAP POINTER.
:POINT POINTER TO FIRST DEVICE.
:CLEAR ERROR COUNT
:CLEAR LAST ERROR POINTER
:SET UP FOR TEST 1
:SET UP FOR POWER FAIL BEFORE
:TESTING STARTS
:HAS INITIALIZATION BEEN PERFORMED
:BR IF YES

```

```

:TYPE TITLE MESSAGE
:IF NOT SET FLAG AND DO
:BIT7=1??
:BR IF NO AUTO SIZE
:GO DO THE AUTO SIZE
;TYPE HEADER
:SET POINTER
:SET DATA
:ALL DONE?
:BR IF YES

:UPDATE POINTER

:IS PROGRAM RUNNING UNDER MONITOR
:BR IF YES
:SELECT SPECIFIC DEVICES??
:BR IF NO.
:TYPE THE MESSAGE.
:ZERO DATA LIGHTS

```

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 11
 CZDVDC.P11 02-FEB-78 13:55 PROGRAM INITIALIZATION AND START UP.

SEQ 0021

```

426 002244 000000          HALT
427 002246 127737 176730 001302  CMPB 2SWR, SAVACT
428 002254 101404          BLOS 2$
429 002256 104402 005243          TYPE ,MERR3
430 002262 000000          HALT
431 002264 000776          BR -2
432 002266 117737 176710 001300 2$: MOV 2SWR, DVACTV
433 002274 113700 001300          MOV  DVACTV, R0
434 002300 042700 177400          BIC  #1C(377), R0
435 002304 000000          HALT
436 002306 012700 000300          3$: MOV  #300, R0
437 002312 012701 000302          MOV  #302, R1
438 002316 010120          4$: MOV  R1, (R0)+
439 002320 005021          CLR  (R1)+
440 002322 022021          CMP  (R0)+, (R1)+
441 002324 022700 001000          CMP  #1000, R0
442 002330 001372          BNE  4$
443
444          ; TEST START AND RESTART
445          ;-----
446
447 002332 012737 000340 177776 .BEGIN: MOV  #340, PS
448 002340 012706 001200          MOV  #STACK, SP
449 002344 005737 000042          TST  2#42
450 002350 001023          BNE  3$
451 002352 032777 000004 176622          BIT  #BIT2, 2SWR
452 002360 001411          BEQ  1$
453 002362 104402 005301          TYPE ,MLOCK
454 002366 012737 000240 002702          MOV  #NOP, TTST
455 002374 012737 000240 002704          MOV  #NOP, TTST+2
456 002402 000406          BR  2$
457 002404 013737 003014 002702 1$: MOV  BRW, TTST
458 002412 013737 003016 002704          MOV  BRX, TTST+2
459 002420          2$:
460 002420 012737 005666 001214 3$: MOV  #CYCLE, RETURN
461 002426 104402 005171          TYPE ,MR
462 002432 000177 176556          JMP  2RETURN

```

```

; WAIT FOR USER TO TELL WHAT DEVICES TO RUN
; IS THE NUMBER VALID?
; BR IF NUMBER IS OK.
; TELL USER OF INVALID NUMBER.
; STOP EVERYTHING.
; RESTART THE PROGRAM AGAIN.
; GET NEW DEVICE PATTERN
; SHOW THE USER WHAT HE SELECTED.
; USE ONLY LOW BYTE.
; CONTINUE DYNAMIC SWITCHES.
; PREPARE TO CLEAR THE FLOATING
; VECTOR AREA. 300-776
; START PUTTING "PC+2 - HALT"
; IN VECTOR AREA.
; POP POINTERS
; ALL DONE??
; BR IF NO.

```

```

; LOCK OUT INTERRUPTS
; SET UP STACK
; IS PROGRAM UNDER MONITOR CONTROL
; BR IF YES
; CHECK FOR LOCK ON TEST
; BR IF NO LOCK DESIRED.
; TYPE LOCK SELECTED.
; ADJUST SCOPE ROUTINE.
; SET UP TO LOCK
; CONTINUE ALONG.
; PREPARE NORMAL SCOPE ROUTINE
; LOCK NOT SELECTED. SET UP FOR NORMAL SCOPE LOOP
; START AT "CYCLE" FIND WHICH DEVICE TO TEST
; TYPE R
; 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      CLRB     ERRFLG      ;CLEAR ERROR FLAG
472      002450 005237 001230      INC      PASCNT      ;UPDATE PASS COUNT
473      002454 013777 001230 176516  MOV     PASCNT,ALIGHTS ;DISPLAY PASS COUNT
474      002462 104402 005145      TYPE     ,MEPASS      ;TYPE END PASS
475      002466 104402 005330      TYPE     ,MCSR      ;TYPE CSR
476      002472 104411 002604      CNVRT    ,XCSR      ;SHOW IT
477      002476 104402 005336      TYPE     ,MVEC      ;TYPE VECTOR
478      002502 104411 002612      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      RESTR      ;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      RESTR      ;IF NOT, CONTINUE TESTING
489      002556 000005      RESET      ;STOP THE SHOW--CLEAR THE WORLD
490
491      002560      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  RESTR:  MOV     #CYCLE,RETURN
497      002600 000137 005666      JMP      CYCLE
498      002604 000001      XCSR:  1
499      002606 006      .BYTE 6,2
500      002610 001362      DVSCR
501      002612 000001      XVEC:  1
502      002614 003      .BYTE 3,2
503      002616 001352      DVRVEC
504      002620 000001      XPASS:  1
505      002622 006      .BYTE 6,2
506      002624 001230      PASCNT
507      002626 000001      XERR:  1
508      002630 006      .BYTE 6,2
509      002632 001232      ERRCNT
510
511      ;SCOPE LOOP AND INTERATION HANDLER
512      ;-----
513
514      002634      .SCOPE:  CMP     #177570,SWR      ;IS THERE A REAL SWR?
515      002634 022737 177570 001202  BEQ     64$      ;BR IF YES
516      002642 001411      MOV     @TKDBR,-(SP)      ;SAVE KEYBOARD CHAR
517      002644 017746 176336      BIC     #BIT7,(SP) ;CLEAR PARITY BIT
518      002650 042716 000200

```

```

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$: CLR LSTERR ; CLEAR LAST ERROR PC.
523 002672 010016 MOV RO, (SP) ; SAVE RO ON THE STACK
524 002674 032777 040000 176300 TTST: BIT #BIT14, QSWR ; "LOOP ON THIS TEST"?
525 002702 001407 BEQ 1$ ; 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 002706 105777 176272 TSTB JTKCSR ; KEYBOARD DONE?
528 002712 100034 BPL 3$ ; BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
529 002714 017700 176266 MOV JTKDBR, RO ; CLEAR DONE BIT
530 002720 000415 BR 2$ ; CONTINUE
531 002722 032777 004000 176252 1$: BIT #SW11, QSWR ; DELETE ITERATION? (QUICK PASS)
532 002730 001011 BNE 2$ ; BR IF YES
533 002732 105737 001313 TSTB QV.FLG ; HAVE PASSES BEEN COMPLETED?
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$: CLRB ERRFLG ; PREPARE FOR NEW TEST
539 002760 005037 001224 CLR LPCNT ; START ICOUNTER AT 0
540 002764 005037 001220 CLR LOCK
541 002770 012737 000024 001222 MOV #20, ICOUNT ; RESET ITERATIONS
542 002776 013737 001216 001214 MOV NEXT, RETURN ; GET NEXT TEST
543 003004 011600 3$: MOV (SP), RO ; POP RO OFF OF THE STACK
544 003006 022626 POP2SP ; FAKE AN "RTI"
545 003010 000177 176200 JMP QRETURN ; GO DO THE TEST
546 003014 001407 BRW: 1407
547 003016 000437 BRX: 437

; CHECK FOR FREEZE ON CURRENT DATA
; -----
550
551
552 003020 032777 001000 176154 .SCOP1: BIT #SW09, QSWR ; 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.

; TELETYPE OUTPUT ROUTINE
; -----
558
559
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$: BIT #SW12, QSWR ; INHIBIT ALL PRINT OUT??
566 003066 001012 BNE 3$ ; BR IF NO PRINT OUT WANTED (SW12=1)
567 003070 105715 TSTB (R5) ; IS NUMBER MINUS? (MSB=1 (BIT?))
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$: TSTB JTPCSR ; TTY READY?
571 003104 100375 BPL 2$ ; BR IF NO.
572 003106 112577 176100 MOVB (R5)+, JTPDBR ; 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 24(SP), .MSG
581 003132 062766 000C02 000004 ADD #2, 4(SP)
582 003140 104402 .INST1: TYPE
583 003142 000000 .MSG: 0
584 003144 012704 005520 MOV #INBUF, R4
585 003150 012703 000007 MOV #7, R3
586 003154 105777 176024 1$: TSTB 2TKCSR
587 003160 100375 BPL 1$
588 003162 117714 176020 MOVB 2TKDBR, (R4)
589 003166 142714 000200 BICB #200, (R4)
590 003172 122427 000015 CMPB (R4), #15
591 003176 001417 BEQ INSTR2
592 003200 105777 176004 2$: TSTB 2TPCSR
593 003204 100375 BPL 2$
594 003206 017777 175774 175776 MOV 2TKDBR, 2TPDBR
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 MQM
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 010546 .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 MOV (R5)+, LOBITS
617 003274 112537 003443 MOV (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 001406 BEQ LIMITS

```

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 15
CZDVDC.P11 02-FEB-78 13:55

SEQ 0025

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

```

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      BLO      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  1$:      MOV      DEVADR,R4
651 003412 010524      MOV      R5,(R4)+
652 003414 062705 000002  ADD      #2,R5
653 003420 105337 003443  DECB     ADRCNT
654 003424 001372      BNE      1$
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 001266      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 013700 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

```

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 16
 CZDVDC.P11 02-FEB-78 13:55 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```

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$: MOV (R1)+,CHRCNT
702 003576 112137 003745 MOV (R1)+,SPACNT
703 003602 013137 003746 MOV @12(SP),BINWRD
704 003606 013704 003746 2$: MOV BINWRD,R4
705 003612 113705 003744 MOV CHRCNT,R5
706 003616 012700 005562 MOV #TEMP,R0
707 003622 010403 3$: MOV R4,R3
708 003624 042703 177770 BIC #177770,R3
709 003630 062703 000060 ADD #060,R3
710 003634 110320 MOV R3,(R0)+
711 003636 000241 CLC
712 003640 006004 ROR R4
713 003642 000241 CLC
714 003644 006004 ROR R4
715 003646 000241 CLC
716 003650 006004 ROR R4
717 003652 005305 DEC R5
718 003654 001362 BNE 3$
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$: CLRB (R3)
729 003714 104402 005624 TYPE ,MDATA
730 003720 005337 003742 DEC WRDCNT
731 003724 001322 BNE 1$
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 003745 003745 SPACNT=CHRCNT+1
741 003746 000000 BINWRD: 0
742

```

```

743
744
745
746
747
748
749 003750 011646
750 003752 162716 000002
751 003756 017616 000000
752 003762 006316
753 003764 042716 177001
754 003770 062716 001314
755 003774 017616 000000
756 004000 000136
757
758
759
760
761 004002
762 004002 022737 177570 001202
763 004010 001411
764 004012 017746 175170
765 004016 042716 000200
766 004022 122726 000007
767 004026 001002
768 004030 004737 004640
769 004034 032777 010000 175140 64$:
770 004042 001406
771 004044 105777 175140
772 004050 100003
773 004052 112777 000207 175132
774 004060 032777 020000 175114 XB$:
775 004066 001105
776 004070 021637 001234
777 004074 001404
778 004076 011637 001234
779 004102 105037 001311
780 004106 104406
781 004110 011605
782 004112 162705 000002
783 004116 011504
784 004120 006304
785 004122 061504
786 004124 006304
787 004126 042704 177001
788 004132 062704 034404
789 004136 012437 004252
790 004142 012437 004264
791 004146 011437 004276
792 004152 105737 001311
793 004156 001403
794 004160 005737 004276
795 004164 001040
796 004166 104402 005104
797 004172 104402 005104
798 004176 005737 001220

; TRAP DISPATCH SERVICE
; ARGUMENT OF TRAP IS EXTRACTED
; AND USED AS OFFSET TO OBTAIN POINTER
; TO SELECTED SUBROUTINE

.TRPSR: MOV (SP), -(SP) ; GET PC OF RETURN
SUB #2, (SP) ; =PC OF TRAP
MOV @2(SP), (SP) ; GET TRP
TRPOK: ASL (SP) ; MULTIPLY TRAP ARG BY 2
BIC #177001, (SP) ; CLEAR UNWANTED BITS
ADD #.TRPTAB, (SP) ; POINTER TO SUBROUTINE ADDRESS
MOV @2(SP), (SP) ; SUBROUTINE ADDRESS
JMP @2(SP)+ ; GO TO SUBROUTINE

; ERROR HANDLER
;-----

.HLT: CMP #177570, -WR ; IS THERE A REAL SWR?
BEQ 64$ ; BR IF YES
MOV @TKDBR, -(SP) ; SAVE KEYBOARD CHAR
BIC #BIT7, (SP) ; CLEAR PARITY BIT
CMPB #7, (SP)+ ; WAS IT CNTRL 'G' ?
BNE .+6 ; BR IF NO.
JSR PC, SERV.G ; SERVICE "CNTRL 'G'".
BIT #SW12, @SWR ; BELL ON ERROR?
BEQ XB$ ; BR IF NO BELL
TSTB @TPCSR ; TTY READY.
BPL XB$ ; DON'T WAIT IF TTY NOT READY.
MOVB #207, @TPDBR ; PUSH A BELL AT THE TTY.
BIT #SW13, @SWR ; DELETE ERROR PRINT OUT?
BNE HALTS ; BR IF NO PRINT OUT WANTED.
CMP (SP), LSTERR ; WAS THIS ERROR FOUND LAST TIME?
BEQ 1$ ; BR IF YES
MOV (SP), LSTERR ; RECORD BEING HERE
CLRB ERRFLG ; PREPARE HEADER
1$: SAVOS ; SAVE ALL PROC REGISTERS
MOV (SP), R5 ; GET THE PC OF ERROR
SUB #2, R5 ; GET ADDRESS OF TRAP CALL
MOV (R5), R4 ; GET HLT INSTRUCTION
ASL R4 ; MULT BY TWO
ADD (R5), R4 ; DOUBLE IT
ASL R4 ; MULT AGAIN
BIC #177001, R4 ; CLEAR JUNK
ADD #.ERRTAB, R4 ; GET POINTER
MOV (R4)+, ERRMSG ; GET ERROR MESSAGE
MOV (R4)+, DATAHD ; GET DATA HEADER
MOV (R4), DATABP ; GET DATA TABLE
TSTB ERRFLG ; TYPE HEADREER
BEQ TYPMSG ; BR IF YES
TST DATABP ; DOES DATA TABLE EXIST?
BNE TYPDAT ; BR IF YES.
TYPMSG: TYPE , MCRLF
TYPE , MCRLF
TST LOCK

```

```

799 004203 001402      BEQ      1$
800 004204 104402 005400      TYPE  ,MASTEK
801 004210 104402 005366      TYPE  ,MTSTN
802 004214 104411 004374      CNVRT  ,XTSTN      ;SHOW IT
803 004220 104402 005454      TYPE  ,MERRPC      ;TYPE PC.
804 004224 104411 004366      CNVRT  ,ERTABO      ;SHOW IT
805 004230 104402 005104      TYPE  ,MCKLF      ;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
810 004252 000000      ERRMSG: 0      ;TYPE
811 004254      WRKO.FM:      ;ERROR MESSAGE
812 004254 005737 004264      TST     DATAHD      ;DATA HEADER?
813 004260 001402      BEQ      TYPDAT      ;BR IF NO
814 004262 104402      TYPE
815 004264 000000      DATAHD: 0      ;TYPE
816 004266 005737 004276      TYPDAT: TST     DATABP      ;DATA HEADER
817 004272 001402      BEQ      RESREG      ;DATA TABLE?
818 004274 104410      CNVRT
819 004276 000300      DATABP: 0      ;BR IF NO.
820 004300 104407      RESREG: RESOS      ;SHOW
821 004302 005777 174674      HALTS: TST     2SWR      ;DATA TABLE
822 004306 100005      BPL      EXITER      ;RESTORE PROC REGISTERS
823 004310 010046      PUSHRO      ;HALT ON ERROR?
824 004312 016600 000002      MOV     2(SP),RO      ;BR IF NO HALT ON ERROR
825 004316 000000      HALT
826 004320 012600      POPRO      ;SAVE RO
827 004322 005237 001232      INC     ERRCNT      ;SHOW ERROR PC IN DATA LIGHTS
828 004326 032777 000400 174646      BIT     8SW08,2SWR      ;HALT
829 004334 001007      BNE      1$      ;GET RO
830 004336 032777 002000 174636      BIT     8SW10,2SWR      ;UPDATE ERROR COUNT
831 004344 001407      BEQ      2$      ;GOTO TOP OF TEST?
832 004346 013737 001216 001214      MOV     NEXT,RETURN      ;BR IF YES
833 004354 012706 001200      1$: MOV     8STACK,SP      ;GOTO NEXT TEST?
834 004360 000177 174630      JMP     2$RETJRN      ;BR IF NO
835 004364 000002      2$: RTI      ;SET FOR NEXT TEST
836 004366 000001      ERTABO: 1      ;RESET SP
837 004370 006      002      ;GOTO SPECIFIED TEST
838 004372 001276      XTSTN: 1      ;RETURN
839 004374 000001      ;.BYTE 3,2
840 004376 003      002      ;TSTNO
841 004400 001226      ;ENTER HERE ON POWER FAILURE
842
843
844
845
846 004402      .PFAIL:
847 004402 012737 004414 000024      MOV     8RESTART,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     8.PFAIL,24      ;SET UP FOR POWER FAILURE

```

CZDVDC0 MACY11 30A(1052) 10-FEB-78 15:44 PAGE 19
 CZDVDC.P11 02-FEB-78 13:55 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```

855 004422 012706 001200      MOV      #STACK,SP      ;RESET THE STACK POINTER
856 004426 005037 005562      CLR      TEMP          ;READY FOR TIMMER
857 004432 005237 005562      INC      TEMP          ;PLUS ONE TO THE TIMER!
858 004436 001375 005562      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 001234      MSTCLR   RAMCLR      ;START CLEAN UP OF DEVICE
864 004462 104413 001234      RAMCLR   ;CLEAR IT ALL!
865 004464 000177 174524      JMP      @RETURN      ;START DOING THAT TEST AGAIN.
866 004470 000001 002        PFTAB: 1
867 004472 003          .BYTE 3,2
868 004474 001226 002        .TSTNO
869 004476 010046 004514      .DELAY: MOV      RO,-(SP)
870 004500 013700 004514      MOV      1$,RO
871 004504 005300 004514      DEC      RO
872 004506 001376 004514      BNE      .-2
873 004510 012600 004514      MOV      (SP)+,RO
874 004512 000002 004514      RTI
875 004514 000036 004514      1$:      30.
876
877 004516 012777 004000 174636 .RAMCLR: MOV      #MRESET,@DVSCR ;ISSUE A MASTER CLEAR
878 004516 010146 004000 174636 MOV      R1,-(SP) ;SAVE R1 ON THE STACK
879 004524 010146 004000 174636 MOV      R4,-(SP) ;SAVE R4 ON THE STACK
880 004526 010446 004000 174636 MOV      DVSR5,R1 ;GET SECONDARY SEL. REG.
881 004530 013701 001372 004000 174636 MOV      DVSR5,R4 ;GET SECONDARY REGISTER ACCESS REG.
882 004534 013704 001376 004000 174636 CLR      (R4) ;ZERO THE SECONDARY REGISTER.
883 004540 005014 004000 174636 ADD      #1<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BITC,(R1)
884 004542 062711 170361 004000 174636 BNE      1$
885 004546 001374 004000 174636 MOV      (SP)+,R4 ;RESTORE R4
886 004550 012604 004000 174636 MOV      (SP)+,R1 ;RESTORE R1
887 004552 012601 004000 174636 RTI
888 004554 000002 004000 174636
889
890 004556 012777 004000 174576 .MSTCLR: MOV      #MRESET,@DVSCR ;ISSUE MASTER CLEAR.
891 004556 012777 004000 174576 RTI
892 004564 000002 004000 174576
893
894 004566 052777 000002 174566 .ROMCLK: BIS      #BIT1,@DVSCR
895 004566 052777 000002 174566 RTI
896 004574 000002 000002 174566
897
898 004576 010046 000400 174560 .DATACLK: MOV      RO,-(SP)
899 004576 005000 000400 174560 CLR      RO
900 004600 005000 000400 174560 BIS      #BIT8,@DVLCA
901 004602 052777 000400 174560 MOV      @DVLCA,3$
902 004610 017737 174554 004636 1$:  MOV      3$+1
903 004616 106037 004637 004636 RORB     2$
904 004622 103003 004637 004636 BCC      RO
905 004624 005200 004637 004636 INC      RO
906 004626 001370 004637 004636 BNE      1$
907 004630 104000 004637 004636 HLT
908 004632 012600 004637 004636 2$:  MOV      (SP)+,RO
909 004634 000002 004637 004636 RTI
910 004636 000001 004637 004636 3$:  .BLKW 1

```

```

911
912 004640 032777 004000 174336 SERV.G: BIT #4000,ATKCSR ;RX BUSY?
913 004646 001374 BNE SERV.G ;ER IF YES
914 004650 017737 174326 005072 174316 1$: MOV @SWR,90$ ;SAVE (SWR).
915 004656 013777 005072 174316 1$: MOV 90$,@SWR
916 004664 104402 005052 TYPE ,89$
917 004670 104411 005064 CNVRT ,88$
918 004674 104402 005074 TYPE ,91$
919 004700 105777 174300 TSTB @TKCSR ;WAIT FOR DONE.
920 004704 100375 BPL -4
921 004706 017746 174274 MOV @TKDBR,-(SP)
922 004712 042716 000200 BIC #BIT7,(SP)
923 004716 122726 000015 CMPB #15,(SP)+
924 004722 001450 BEQ 5$
925 004724 005077 174252 CLR @SWR
926 004730 105777 174254 2$: TSTB @TPCSR
927 004734 100375 BPL -4
928 004736 016677 177776 174246 MOV -2(SP),@TPDBR
929 004744 000241 CLC
930 004746 006177 174230 ROL @SWR
931 004752 006177 174224 ROL @SWR
932 004756 006177 174220 ROL @SWR
933 004762 103735 BCS 1$ ;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 #1C<7>,-2(SP)
939 005012 056677 177776 174162 BIS -2(SP),@SWR
940 005020 105777 174160 TSTB @TKCSR
941 005024 100375 BPL -4
942 005026 017746 174154 MOV @TKDBR,-(SP)
943 005032 042716 000200 BIC #BIT7,(SP)
944 005036 122726 000015 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
951
952 005064 000001 .EVEN
953 005066 006 000 88$: 1
954 005070 005072 .BYTE 6.0
955 005072 000000 90$: .WORD 0
956 005074 036457 000057 91$: .ASCIZ ?/=/?
957
958 005100 020040 000077 .EVEN
959 (2) 005104 005015 000 MQM: .ASCIZ / ?/
960 (2) 005107 377 053520 020122 MCRLF: .ASCIZ <15><12>
961 (2) 005145 377 047105 020104 MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST /
962 (2) 005171 377 000122 MEPASS: .ASCIZ <377>/END PASS CZDVDCO /
963 (2) 005174 050377 047522 051107 MR: .ASCIZ <377>/R/
964 (2) 005243 377 047111 052523 MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
965 (2) 005267 377 042524 052123 MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
966 (2) 005301 377 047514 045503 MTSTPC: .ASCIZ <377>/TEST PC-/
967 MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/

```

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 21
 CZDVDC.P11 02-FEB-78 13:55 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

(2)	005330	051503	035122	000040	MCSR: .ASCIZ /CSR: /
(2)	005336	042526	035103	000040	MVEC: .ASCIZ /VEC: /
(2)	005344	040520	051523	051505	MPASS: .ASCIZ /PASSES: /
(2)	005355	105	051122	051117	MERR: .ASCIZ /ERRORS: /
(2)	005366	042524	052123	047040	MTSTN: .ASCIZ /TEST NO: /
(2)	005400	000052			MASTEK: .ASCIZ /*/
(2)	005402	051777	052105	051440	MNEW: .ASCIZ <37> /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-OUTPLT
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
975
976
977
978
979
980
981
982 005666 105737 001300      CYCLE:  TSTB  DVACTV  ;ARE ANY DV11'S TO BE TESTED?
983 005672 001004      BNE  1$      ;BR IF OK.
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 DV1:
997 005754 000241      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      MOV  (RO)+,DVSCR  ;RESTORE POINTER.
1007 006022 012037 001352      MOV  (RO)+,DVRVEC ;LOAD SYSTEM CTRL. REG
1008 006026 012037 001416      MOV  (RO)+,LO0.03 ;LOAD VECTOR
1009 006032 012037 001426      MOV  (RO)+,SYNC2A ;GET LINE PARAMETERS. 00-03
1010 006036 012037 001420      MOV  (RO)+,LO4.07 ;
1011 006042 012037 001430      MOV  (RO)+,SYNC2B ;
1012 006046 012037 001422      MOV  (RO)+,LO8.11 ;
1013 006052 012037 001432      MOV  (RO)+,SYNC2C ;
1014 006056 012037 001424      MOV  (RO)+,L12.15 ;
1015 006062 012037 001434      MOV  (RO)+,SYNC2D ;
1016 006066 012700 000002      MOV  #2,RO      ;SAVE CORE THIS WAY!
1017 006072 013737 001364 001364      MOV  DVSCR,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,CVLCR ;GET LN. PAR.REG.
1022 006124 060037 001370      ADD  RO,DVLCR ;GOT IT
1023 006130 013737 001370 001372      MOV  DVLCR,DVSRS ;GET SEC. REG. SEL. REG.
1024 006136 060037 001372      ADD  RO,DVSRS ;GOT IT
1025 006142 013737 001372 001374      MOV  DVSRS,DVSRSH ;GET HIGH BYTE.
1026 006150 005237 001374      INC  DVSRSH ;GOT IT
1027 006154 013737 001374 001376      MOV  DVSRSH,DVSRA ;SEC. REG. ACCESS.
1028 006162 005237 001376      INC  DVSRA      ;GOT IT

```

H03

SEQ 0033

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 23

CZDVDC.P11 02-FEB-78 13:55

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

1029	006166	013737	001376	001400	MOV	DVSRA,DVSFR	;SPEC. FUN. REG.
1030	006174	060037	001400		ADD	RO,DVSFR	
1031	006200	013737	001400	001402	MOV	DVSFR,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	DVRVEC,DVRLVL	;PTY LVL
1037	006232	060037	001354		ADD	RO,DVRLVL	
1038	006236	013737	001354	001356	MOV	DVRLVL,DVTEVC	;TX VEC
1039	006244	060037	001356		ADD	RO,DVTEVC	
1040	006250	013737	001356	001360	MOV	DVTEVC,DVTLVL	;TX LVL
1041	006256	060037	001360		ADD	RO,DVTLVL	
1042							
1043	006262	012700	001416		MOV	#L00.03,RO	;LOAD STAU 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 STAU 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 STAU 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 STAU 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			BEG	7\$	
1064	006372						
1065	006372	005737	000042		TST	2#42	
1066	006376	001042			BNE	7\$	
1067	006400	104402	005104		TYPE	,MCRLF	
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			0		
1075	006421	001			1		
1076	006422	012700	007256		MOV	#TST1,RO	
1077	006426	022710			CMP	(PC)+,(RO)	
1078	006430	012737			MOV	(PC)+,2(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          6$
1087 006466 005720          6$:      TST      (R0)+
1088 006470 020027 021152          CMP      R0, #TLAST+10
1089 006474 001354          BNE      5$
1090 006476 104402 005100          TYPE      MQM
1091 006502 000733          BR          4$
1092 006504 012737 007256 001214 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      #1C<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 105011          CLRB      (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$:      RTS      PC
1118
1119          ; *ROUTINE USED TO "AUTO SIZE" THE DV11
1120          ; *CSR AND VECTOR.
1121          ; *NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
1122          ; *      ADDRESS RANGE (175000:175400)
1123          ; *      AND THE VECTOR MAY BE ANY WHERE IN THE
1124          ; *      FLOATING VECTOR RANGE (300:770)
1125          ; *
1126          ; *
1127 006622          AUTO.SIZE:
1128 006622 000005          RESET
1129 006624 012702 001500          CSRMAP: MOV      #DV.MAP, R2      ; INSURE A BUS INIT.
1130 006630 005022          1$:      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          CLRB      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
1136 006654 012737 007074 000004          2$:      MOV      #6$, #4      ; SET FOR FIRST ADDRESS TO BE TESTED
1137 006662 005711          TST      (R1)      ; SET FOR NON-EXISTANT DEVICE TIME OUT
1138 006664 001037          BNE      3$      ; IF DV11 DVSCR S/B 0
1139 006666 022761 177777 000012          CMP      #177777, 12(R1)      ; IF NO DEV; TRAP TO 4. IF NO BIT 8 THEN NO DV11
1140 006674 001033          BNE      3$      ; IF DV11 THEN DVSCR S/B ALL 1'S ON INIT!
          ; 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 070226      BIS      #226, (R2)+      ; SET LINE CARD 1 STAT AND SYNC
1147 006714 052722 170062      BIS      #62, (R2)+      ;
1148 006720 052722 0J0226      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 001301      INCB     DVNUM      ; UPDATE DEVICE COUNTER
1155 006754 122737 000010 001301  CMPB     #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$:      ADD      #10, R1      ; UPLD CSR POINTER ADDRESS
1158 006770 022701 175400      CMP      #175400, R1
1159 006774 001332      BNE      2$          ; BR IF MORE ADDRESS TO CHECK.
1160 006776 012722 177777 100$:      MOV      #177777, (R2)+      ; TERMINATER.
1161 007002 105037 001300      CLRB     DVACTV
1162 007006 105737 001301      TSTB     DVNUM      ; WERE ANY DV11'S FOUND AT ALL?
1163 007012 001423      BEQ      5$          ; ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
1164 007014 113701 001301      MOVB     DVNUM, R1
1165 007020 110137 001303      MOVB     R1, SAVNUM      ; SAVE NUMBER OF DEVICES
1166 007024 000241 4$:      CLC
1167 007026 106137 001300      ROLB     DVACTV      ; GENERATE ACTIVE REGISTER OF DEVICES.
1168 007032 105237 001300      INCB     DVACTV      ; SET THE BIT
1169 007036 005301      DEC      R1
1170 007040 001371      BNE      4$          ; BR IF MORE TO GENERATE
1171 007042 012737 000006 000004  MOV      #6, #4      ; RESTORE TRAP VECTOR
1172 007050 113737 001300 001302  MOVB     DVACTV, SAVACT ; SAVE ACTIVE REGISTER
1173 007056 000137 007102      JMP      VECMAP      ; GO FIND THE VECTOR NOW.
1174 007062 104402 005174 5$:      TYPE     MERR2      ; NOTIFY OPR THAT NO DV11'S FOUND.
1175 007066 005000      CLR      R0      ; MAKE DATA LIGHTS ZERO
1176 007070 000000      HALT
1177 007072 000776      BR      -2          ; STOP THE SHOW
1178 007074 012716 006764 6$:      MOV      #35, (SP)      ; DISABLE CONT. SW.
1179 007100 000002      RTI          ; ENTERED BY NON-EXISTANT TIME-OUT.
1180      ; RETURN TO MAINSTREAM
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, R0      ; FLOATING VECTORS START HERE.
1185 007126 012701 000302      MOV      #302, R1      ; PC OF IOT INSTR.
1186 007132 010120 1$:      MOV      R1, (R0)+      ; START FILLING VECTOR AREA
1187 007134 012721 000004      MOV      #4, (R1)+      ; WITH .+2; IOT
1188 007140 022021      CMP      (R0)+, (R1)+      ; ADD 2 TO R0 +R1
1189 007142 020127 001000      CMP      R1, #1000
1190 007146 101771 1$:      BLOS     1$          ; BR IF MORE TO FILL
1191 007150 113737 001300 001246  MOVB     DVACTV, TEMP1 ; STORE TEMPORALLY
1192 007156 006037 001246 2$:      ROR      TEMP1      ; BRING OUT A BIT
1193 007162 103034      BCC      5$          ; BR IF ALL DONE
1194 007164 005037 177776      CLR      PS      ; ZERO CPU PRIO
1195 007170 012772 001300 000000  MOV      #BIT9+BIT7+BIT6, #2(R2)
1196 007176 005000      CLR      R0      ; ATTEMPT TO FORCE AN INTERRUPT

```

CZDVDCO MACY11 30A(1052) 15-FEB-78 15:44 PAGE 26
CZDVDC.P11 02-FEB-78 13:55 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

1197	007200	005200				INC	R0	;STALL
1198	007202	001376				BNE	.-2	; FOR TIME TO INTERRUPT
1199	007204	052762	000300	000002		BIS	#300,2(R2)	;NO INTERRUPT ASSUME 300 AND FIX D/J1 LATER
1200	007212	042772	176777	000000	3\$:	BIC	#1C(BIT9),2(R2)	
1201	007220	005072	000000			CLR	2(R2)	
1202	007224	062702	000024			ADD	#24,R2	;POP SOFTWARE POINTER
1203	007230	000752				BR	2\$	;KEEP GOING
1204	007232	051662	000002		4\$:	BIS	(SP),2(R2)	;GET VECTOR ADDRESS
1205	007236	042762	000007	000002		BIC	#7,2(R2)	;CLEAR JUNK
1206	007244	022626				CMP	(SP)+(SP)+	;POP IOT JUNK OFF STACK
1207	007246	012716	007212			MOV	#3\$, (SP)	;SET FOR RETURN
1208	007252	000002				RTI		
1209	007254	000207			5\$:	RTS	PC	;ALL DONE WITH "AUTO SIZING"
1210								

```

1211
1212
1213
1214
1215
1216
1217
1218
1219
1220 007256 012737 000001 001226
1221 007264 012737 007664 001216
1222 007272 012700 000000
1223 007276 013737 001416 001236
1224 007304 100402
1225 007306 004717 007374
1226 007312 012700 000004
1227 007316 013737 001420 001236
1228 007324 100402
1229 007326 004737 007374
1230 007332 012700 000010
1231 007336 013737 001422 001236
1232 007344 100402
1233 007346 004737 007374
1234 007352 012700 000014
1235 007356 013737 001424 001236
1236 007364 100402
1237 007366 004737 007374
1238 007372 104400
1239 007374
1240 007374 012737 007422 001220
1241 007402 104413
1242 007404 005003
1243 007406 005001
1244 007410 112737 000025 022560
1245 007416 012702 000004
1246 007422 110137 023605
1247 007426 010077 171740
1248 007432 004537 022120
1249 007436 000 001
1250 007440 022560
1251 007442 177777
1252 007444 004537 022120
1253 007450 013 010
1254 007452 000004
1255 007454 023560
1256 007456 004537 022120
1257 007462 014 014
1258 007464 000000
1259 007466 000000
1260 007470 032737 004000 001236
1261 007476 001407
1262 007500 004537 022164
1263 007504 015000
1264 007506 004537 022164
1265 007512 072000
1266

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

; TEST 1
-----
1ST1: MOV #1,TSTNO
      MOV #TST2,NEXT
      MOV #0,R0
      MOV L00.03,STAT
      BMI 100$
      JSR PC,105$
100$: MOV #4,R0
      MOV L04.07,STAT
      BMI 101$
      JSR PC,105$
101$: MOV #8,R0
      MOV L08.11,STAT
      BMI 102$
      JSR PC,105$
102$: MOV #12,R0
      MOV L12.15,STAT
      BMI 103$
      JSR PC,105$
103$: SCOPE
105$:

      MOV #1$,LOCK
      RAMCLR
      CLR R3
      CLR R1
      MOV #25,TXBAP
      MOV #4,R2
      MOV R1,XTAB+25
      MOV R0,ADVSR5
      PERFORM SETREG
      .BYTE 000,001
      TXBAP
      -1
      PERFORM SETREG
      .BYTE 013,010
      BIT2
      TXTAB
      PERFORM SETREG
      .BYTE 014,014
      0
      0
      BIT #ASYNC,STAT
      BEQ 60$
      PERFORM LOAD.MODE
      <BIT12+BIT11>+BIT9
      PERFORM LOAD.MODE
      <BIT14+BIT13+BIT12>+BIT10

      PLACE LINE NUMBER INTO R0
      LOAD LINE CARD STATUS INTO STAT
      BR IF LINE CARD NOT TO BE TESTED
      GO DO THE TEST FOR LINE CARD 1
      PLACE LINE NUMBER INTO R0
      LOAD LINE CARD STATUS INTO STAT
      BR IF LINE CARD NOT TO BE TESTED
      GO DO THE TEST FOR LINE CARD 2
      LOAD LINE NUMBER
      LOAD LINE CARD STATUS INTO STAT
      BR IF LINE CARD NOT TO BE TESTED
      DO THE TEST FOR LINE CARD 3
      LOAD LINE NO.
      LOAD LINE CARD STATUS
      BR IF LINE CARD NOT TO BE TESTED
      DO THE TESTS FOR LINE CARD 4
      SCOPE THIS TEST.
      TEST ENTRANCE.
      SET IF SW09=1 (LOCK)
      CLEAR ALL SEC REGISTERS
      SET IMAGE EXPECTED MODE=0
      SET IMAGE "NEXT MODE"=0
      SET TX DATA CHAR
      SET FOR - LINE GROUP
      LOAD CONTROL BYTE(MODE)
      LOAD LINE NUMBER

      TX PRINCIPLE BA, PRINCIPLE BC

      LINE STATE, CNTRL TABLE
      TXGO

      TX MODE REG
      MAKE
      IT=0
      #IS THIS ASYNC LINE CARD?
      #BR IF NO.
      #
      #8 BITS/PER/CHAR
      #
      ;#9600 BAUD.

```

```

1267 007514 000403      BR      61$
1268 007516 004537 022164 60$:  PERFORM  LOAD.MODE      ;LOAD
1269 007522 014000      BIT12+BIT11 ;MODE
1270 007524 012737 000340 177776 61$:  MOV      #340,PS      ;LOCK OUT INTERRUPTS
1271 007532 012777 007574 171616      MOV      #3$,ADVTEC      ;SET TRANS VECTOR
1272 007540 012777 000340 171612      MOV      #340,ADVTLVL ;LOAD PRIO
1273 007546 052777 020001 171606      BIS      #BIT13+BIT0,ADVSCR ;SET STATUS IE AND UCPU GO.
1274 007554 005005      CLR      R5      ;WAIT
1275 007556 104414      DELAY      ;STALL FOR TIME
1276 007560 005037 177776      CLR      PS      ;ALLOW INTERRUPTS (NSR ENTRY)
1277 007564 005205      INC      R5      ;ENTRY
1278 007566 001373      BNE      2$
1279 007570 104000      HLT
1280 007572 024646      CMP      -(SP),-(SP) ;NO SILO ENTRY (DVSCR IS NOT=1)
1281 007574 042777 020000 171560 3$:  BIC      #BIT13,ADVSCR ;FAKE INTERRUPT BECAUSE NO REAL ONE HAPPENED.
1282 007602 005037 177776      CLR      PS      ;CLR IE
1283 007606 022626      CMP      (SP)+,(SP)+ ;ZERO PSW
1284 007610 112777 000014 171556      MOV      #14,ADVSRSH ;FAKE AN RTI
1285 007616 017704      MOV      ADVSRA,R4 ;SEL TX MODE REGISTER
1286 007622 010305      MOV      R3,R5      ;READ MODE REG.
1287 007624 020504      CMP      R5,R4      ;SET EXPECTED
1288 007626 001401      BEQ      4$      ;WAS "NEXT MODE" LOADED CORRECTLY?
1289 007630 104003      HLT      3      ;BR IF YES
1290 007632 104412      MSTCLR ;TX MODE REGISTER WRONG
1291 007634 104401      SCOP1 ;INIT DV11
1292 007636 005203      INC      R3      ;LOCK ON MODE, LOCK ON LINE?
1293 007640 062701 000040      ADD      #BITS,R1 ;UPDATE EXPECTED MODE
1294 007644 105701      TSTB      R1      ;UPDATE CNTRL BYTE IMAGE
1295 007646 001665      BEQ      1$      ;ALL DONE??
1296 007650 005001      CLR      R1      ;BR IF NO
1297 007652 005003      CLR      R3      ;ZERO EXPECTE MODE
1298 007654 005200      INC      R0      ;ZERO CNTRL BYTE MODE
1299 007656 005302      DEC      R2      ;UPDATE LINE NO POINTER
1300 007660 001260      BNE      1$      ;4 LINES DONE
1301 007662 000207      RTS      PC      ;BR IF YES
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

```

1314 007664 012737 000002 001226 1ST2:  MOV      #2,TSTNO
1315 007672 012737 010430 001216      MOV      #TST3,NEXT
1316 007700 012700 000000      MOV      #0,R0      ;PLACE LINE NUMBER INTO R0
1317 007704 113737 001406 001244      MOV      MASK.A,MASKX ;PLACE "MASK" FOR CHARS INTO MASKX
1318 007712 013737 001416 001236      MOV      LDD.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,R0      ;PLACE LINE NUMBER INTO R0
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	012700	000010		101\$: MOV	#8.,R0	: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		102\$: MOV	#12.,R0	:LOAD LINE NO.
1332	010006	113737	001411	001244	MOV	MASK.D,MASKX	:GET MASKK
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			103\$: SCOPE		:SCOPE THIS TEST.
1337	010032				105\$:		: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	PC	: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			1\$: 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	#1\$,TXBAP+1	:DTA
1352	010112	012702	000004		MOV	#4,R2	:SET FOR 4 LINE GROUP
1353	010116	010077	171250		3\$: MOV	R0,ADVSR5	: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	000	001		.BYTE	000,001	:REGISTERS
1361	010154	022560			TXBAP		:LOAD FOR ASYNC
1362	010156	177776			-2		:LOAD FOR ASYNC
1363	010160	000405			BR	81\$	:CONTINUE TEST
1364	010162	004537	022120		80\$: PERFORM	SETREG	
1365	010166	000	001		.BYTE	000,001	:TX PRINCIPLE BA, PRINCIPLE BC
1366	010170	022556			SYNC		
1367	010172	177774			-4		
1368	010174	004537	022120		81\$: PERFORM	SETREG	
1369	010200	004	005		.BYTE	004,005	:RX BA, RX BC
1370	010202	027560			RXBA		
1371	010204	177772			-6		
1372	010206	004537	022120		PERFORM	SETREG	
1373	010212	010	011		.BYTE	010,011	:TX TABLE, RXTABLE
1374	010214	023560			TXTAB		
1375	010216	030160			RXTAB		
1376	010220	004537	022120		PERFORM	SETREG	
1377	010224	013	012		.BYTE	013,012	:LINE STATE, LINE PROTOCOL
1378	010226	000004			BIT2		:TX GOOD

```

1379 010230 000000 0
1380 010232 032737 004000 001236 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$: PERFORM ,LOAD.MODE ;LOAD
1391 010272 034000 BIT13+BIT12+BIT11 ;MODE AND RX ENABLE
1392 010274 004537 021706 PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
1393 010300 005277 171056 4$: 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=1!
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 6$: MOVB STAT,R5 ;SET EXPECTED=SYNC CHAR
1408 010350 113705 001236 BIC #1<377>,R5 ;CLEAR HIGH BYTE
1409 010354 042705 177400 BICB MASKX,R5 ;CLEAR BITS/PER/CHAR MASK.
1410 010360 143705 001244 MOV #4,R3 ;SET TO LOOK AT 4 CHARS
1411 010364 012703 000004 MOV ,RXBA+2,R1 ;GET RX DATA POINTER
1412 010370 012701 027562 7$: MOVB (R1)+,R4 ;GET FOUND DATA
1413 010374 112104 BIC #1<377>,R4 ;CLEAN HIGH BYTE
1414 010376 042704 177400 CMP R5,R4
1415 010402 020504 BEQ 8$
1416 010404 001401 HLT 1 ;TRANSMITTER IDLED WRONG
1417 010406 104001 8$: DEC R3 ;4 CHARS CHECKED?
1418 010410 005303 BNE 7$ ;BR IF NO
1419 010412 001370 MSTCLR ;INIT DV11
1420 010414 104412 SCOPI ;LOCK ON LINE?
1421 010416 104401 INC R0 ;UPDATE LINE POINTER
1422 010420 005200 DEC R2 ;4 LINE GROUP DONE?
1423 010422 005302 BNE 3$ ;BR IF NO
1424 010424 001234 RTS PC ;EXIT FOR NEXT GROUP
1425 010426 000207
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
1436
1437
1438 010430 012737 000003 001226
1439 010436 012737 011174 001216
1440 010444 012700 000000
1441 010450 113737 001406 001244
1442 010456 013737 001416 001236
1443 010464 100402
1444 010466 004737 010576
1445 010472 012700 000004
1446 010476 113737 001407 001244
1447 010504 013737 001420 001236
1448 010512 100402
1449 010514 004737 010576
1450 010520 012700 000010
1451 010524 113737 001410 001244
1452 010532 013737 001422 001236
1453 010540 100402
1454 010542 004737 010576
1455 010546 012700 000014
1456 010552 113737 001411 001244
1457 010560 013737 001424 001236
1458 010566 100402
1459 010570 004737 010576
1460 010574 104400
1461 010576
1462 010576 032737 004000 001236
1463 010604 001401
1464 010606 000207
1465 010610 012737 010662 001220
1466 010616 104413
1467 010620 012705 023560
1468 010624 012704 030160
1469 010630 005001
1470 010632 005025
1471 010634 005024
1472 010636 105201
1473 010640 100374
1474 010642 012737 000001 022560
1475 010650 112737 000015 022561
1476 010656 012702 000004
1477 010662 010077 170504
1478 010666 005037 027560
1479 010672 005037 027562
1480 010676 005037 027564
1481 010702 032737 004000 001236
1482 010710 001406
1483 010712 004537 022120
1484 010716 000 001
1485 010720 022560
1486 010722 177776
1487 010724 000405
1488 010726 004537 022120
1489 010732 000 001
1490 010734 022556

: TEST 3
-----
153: MOV #3,TSTNO
MOV #TST4,NEXT
MOV #0,R0
MOV# MASK.A,MASKX
MOV LO0.03,STAT
BMI 100$
JSR PC,105$
100$: MOV #4,R0
MOV# MASK.B,MASKX
MOV LO4.07,STAT
BMI 101$
JSR PC,105$
101$: MOV #8,R0
MOV# MASK.C,MASKX
MOV LO8.11,STAT
BMI 102$
JSR PC,105$
102$: MOV #12,R0
MOV# MASK.D,MASKX
MOV L12.15,STAT
BMI 103$
JSR PC,105$
103$: SCOPE
105$: BIT #ASYNC,STAT
BEQ .+4
RTS PC
MOV #3$,LOCK
RAMCLR
MOV #TXTAB,R5
MOV #RXTAB,R4
CLR R1
1$: CLR (R5)+
CLR (R4)+
INCB R1
BPL 1$
MOV #1,TXBAP
MOV# #1$,TXBAP+1
MOV #4,R2
3$: MOV R0,JDVSR5
CLR RXBA
CLR RXBA+2
CLR RXBA+4
BIT #ASYNC,STAT
BEQ 80$
PERFORM SETREG
.BYTE 000,001
TXBAP
-2
BR 81$
80$: PERFORM SETREG
.BYTE 000,001
SYNC

: PLACE LINE NUMBER INTO R0
: PLACE "MASK" FOR CHARS INTO MASKX
: LOAD LINE CARD STATUS INTO STAT
: BR IF LINE CARD NOT TO BE TESTED
: GO DO THE TEST FOR LINE CARD 1
: PLACE LINE NUMBER INTO R0
: GET MASK
: LOAD LINE CARD STATUS INTO STAT
: BR IF LINE CARD NOT TO BE TESTED
: GO DO THE TEST FOR LINE CARD 2
: LOAD LINE NUMBER
: GET MASK
: LOAD LINE CARD STATUS INTO STAT
: BR IF LINE CARD NOT TO BE TESTED
: DO THE TEST FOR LINE CARD 3
: LOAD LINE NO.
: GET MASKK
: LOAD LINE CARD STATUS
: BR IF LINE CARD NOT TO BE TESTED
: DO THE TESTS FOR LINE CARD 4
: SCOPE THIS TEST.
: TEST ENTRANCE.
: IS THIS AN ASYNC LINE CARD?
: BR IF NOT ASYNC.
: EXIT TEST. (ASYNC LINE CARD NOT TESTED)
: SET FOR RETURN IF SW09=1
: CLEAR ALL SEC REGISTERS
: CLEAR
: RECEIVER
: AND
: TRANSMITTER
: CONTROL
: TABLES
:
: LOAD TX
: DTA
: SET FOR 4 LINE GROUP
: LOAD LINE NUMBER
: CLEAR
: RECEIVER
: BUFFER
: IS THIS AN ASYNC LINE CARD?
: BR IF NOT ASYNC.
: ADJUST FOR ASYNC LINE CARD
: #REGISTERS
: #LOAD FOR ASYNC
: #LOAD FOR ASYNC
: #CONTINUE TEST
: TX PRINCIPLE BA, PRINCIPLE BC
:

```

1491	010736	177774			-4		
1492	010740	004537	022120	81\$:	PERFORM	SETREG	
1493	010744	004	005		BYTE	004,005	RX BA, RX BC
1494	010746	027560			AXBA		
1495	010750	177772			-6		
1496	010752	004537	022120		PERFORM	SETREG	
1497	010756	010	011		BYTE	010,011	TX TABLE, RXTABLE
1498	010760	023560			TXTAB		
1499	010762	030160			RXTAB		
1500	010764	004537	022120		PERFORM	SETREG	
1501	010770	013	012		BYTE	013,012	LINE STATE, LINE PROTOCOL
1502	010772	000004			BIT2		TX GOOD
1503	010774	000001			BIT0		IDLE MARK ON BYTE CNT=0
1504	010776	032737	004000 001236		BIT	#ASYNC, STAT	#IS THIS ASYNC LINE CARR?
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		#8 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	170306		TSB	DVSCR	WAIT FOR
1519	011054	100375			BPL	-4	DVSCR07=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=1!
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			6\$:			
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		7\$:	MOVB	(R1)+, R4	GET FOUND DATA
1538	011142	042704	177400		BIC	#C<377>, R4	CLEAR 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			SCOPI		LOCK ON LINE?
1546	011164	005200			INC	RO	UPDATE LINE POINTER

1547 011166 005302
1548 011170 001234
1549 011172 000207

DEC R2
BNE 35
RTS PC

;4 LINE GROUP DONE?
;BR IF NO
;EXIT FOR NEXT GROUP

```

***** TEST 4 *****
;TEST OF RECEIVER CONTROL BYTE OPERATIONS.
;TEST OF THE "STORE/DISCARD" FUNCTIONS.
;TEST THAT CHARS:
; 25 STORED
; 23 DISCARDED
; 31 STORED
; 32 DISCARDED
;SINCE TWO CHRS SHOULD BE THROWN AWAY:
;THE TX LINE IS SET TO GO BACK TO A MARK STATE;
;THEREFORE THE RX BUFFER S/B:
;RXBA 31,25
; 377,377
;(AT 8 BITS PER CHAR)
;THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
*****

```

; TEST 4

1571 011174 012737 000004 001226
1572 011202 012737 012004 001216
1573 011210 012700 000000
1574 011214 113737 001406 001244
1575 011222 013737 001416 001236
1576 011230 100402
1577 011232 004737 011342
1578 011236 012700 000004
1579 011242 113737 001407 001244
1580 011250 013737 001420 001236
1581 011256 100402
1582 011260 004737 011342
1583 011264 012700 000010
1584 011270 113737 001410 001244
1585 011276 013737 001422 001236
1586 011304 100402
1587 011306 004737 011342
1588 011312 012700 000014
1589 011316 113737 001411 001244
1590 011324 013737 001424 001236
1591 011332 100402
1592 011334 004737 011342
1593 011340 104400
1594 011342
1595 011342 012737 011442 001220
1596 011350 104413
1597 011352 105037 023605
1598 011356 105037 023603
1599 011362 105037 023611
1600 011366 105037 023612
1601 011372 105037 024157
1602 011376 012705 022560

```

;-----
;ST4: MOV #4,TSTNO
      MOV #TSTS,NEXT
      MOV #0,R0
      MOV MASK.A,MASKX
      MOV LO0.03,STAT
      BMI 100$
      JSR PC,105$
100$: MOV #4,R0
      MOV MASK.B,MASKX
      MOV LO4.07,STAT
      BMI 101$
      JSR PC,105$
101$: MOV #8,R0
      MOV MASK.C,MASKX
      MOV LO8.11,STAT
      BMI 102$
      JSR PC,105$
102$: MOV #12,R0
      MOV MASK.D,MASKX
      MOV L12.15,STAT
      BMI 103$
      JSR PC,105$
103$: SCOPE
105$: MOV #15,LOCK
      RAMCLR
      CLRB TXTAB+25
      CLRB TXTAB+23
      CLRB TXTAB+31
      CLRB TXTAB+32
      CLRB TXTAB+377
      MOV #TXBAP,R5

```

```

;PLACE LINE NUMBER INTO R0
;PLACE "MASK" FOR CHARS INTO MASKX
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 1
;PLACE LINE NUMBER INTO R0
;GET MASK
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 2
;LOAD LINE NUMBER
;GET MASK
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TEST FOR LINE CARD 3
;LOAD LINE NO.
;GET MASKK
;LOAD LINE CARD STATUS
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TESTS FOR LINE CARD 4
;SCOPE THIS TEST.
;TEST ENTRANCE.
;SET RETURN IF SW09=1
;CLEAR ALL DV11 SEC REGISTERS
;ZERO
      USED
      CONTROL
      BYTES
;FOR TRANSMITTER
;LOAD

```

1603	011402	012725		MOV	(PC)+,(R5)+	;TRANSMITTER
1604	011404	025	023	.BYTE	25,23	;DATA
1605	011406	012715		MOV	(PC)+,(F5)	;CHARS
1606	011410	031	032	.BYTE	31,32	
1607	011412	112737	000020 030203	MOVB	#BIT4,RXTAB+23	;OSCARC
1608	011420	112737	000020 030212	MOVB	#BIT4,RXTAB+32	;OSCARC
1609	011426	105037	030205	CLRB	RXTAB+25	;DEFAULT-STORE
1610	011432	105037	030211	CLRB	RXTAB+31	;DEFAULT-STORE
1611	011436	012702	000004	MOV	#4,R2	;SET FOR 4 LINE GROUP
1612	011442	010077	167724	MOV	R0,ADVSR5	;LOAD LINE NO.
1613	011446	005037	027560	CLR	RXBA	;MAKE SURE
1614	011452	005037	027562	CLR	RXBA+2	;RX BUFFER=0
1615	011456	032737	004000 001236	BIT	#ASYNC,STAT	;IS THIS AN ASYNC LINE CARD?
1616	011464	001406		BEQ	80\$	;BR IF NOT ASYNC.
1617	011466	004537	022120	PERFORM	SETREG	;ADJUST FOR ASYNC LINE CARD
1618	011472	000	001	.BYTE	000,001	;REGISTERS
1619	011474	022560		TXBAP		;LOAD FOR ASYNC
1620	011476	177774		-4		;LOAD FOR ASYNC
1621	011500	000405		BR	81\$	;CONTINUE TEST
1622	011502	004537	022120	PERFORM	SETREG	
1623	011506	000	001	.BYTE	000,001	;TX PRINCIPLE BA, PPINCIPLE 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	82\$	;BR IF NOT ASYNC.
1629	011524	004537	022120	PERFORM	SETREG	;ADJUST FOR ASYNC LINE CARD
1630	011530	004	005	.BYTE	004,005	;REGISTERS
1631	011532	027560		RXBA		;LOAD FOR ASYNC
1632	011534	177776		-2		;LOAD FOR ASYNC
1633	011536	000405		BR	83\$	;CONTINUE TEST
1634	011540	004537	022120	PERFORM	SETREG	
1635	011544	004	005	.BYTE	004,005	;RXBA, RXBC
1636	011546	027560		RXBA		
1637	011550	177774		-4		
1638	011552	004537	022120	PERFORM	SETREG	
1639	011556	010	011	.BYTE	010,011	;TX TABLE, RX TABLE
1640	011560	023560		TXTAB		
1641	011562	030160		RXTAB		
1642	011564	004537	022120	PERFORM	SETREG	
1643	011570	013	012	.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	60\$	;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	072000		<BIT14+BIT13+BIT12>+BIT10		
1654						;9600 BAUD.
1655	011630	000405		BR	2\$	
1656	011632	004537	022164	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 100404              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      4$: MOV      #25,R5      :SET EXPECTED
1669 011674 113704 027560      MOVB     RXBA,R4      :GET FOUND
1670 011700 020504              CMP      R5,R4      :OK?
1671 011702 001401              BEQ      5$          :
1672 011704 104002              HLT      2          :'25' NOT FIRST IN RX BUFFER
1673 011706 012705 000031      5$: MOV      #31,R5      :NEXT CHAR S/B '31'
1674 011712 113704 027561      MOVB     RXBA+1,R4    :GET NEXT CHAR
1675 011716 120504              CMPB     R5,R4      :OK
1676 011720 001401              BEQ      6$          :
1677 011722 104002              HLT      2          :'31' NOT SECOND IN RX BUFFER
1678 011724 032737 004000 001236 6$: BIT      #ASYNC,STAT :IS THIS AN ASYNC LINE CARD?
1679 011732 001016              BNE      8$          :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      8$: MSTCLR              :INIT DV11
1693 011772 104401              SCOPI              :LOCK ON CURRENT LINE?
1694 011774 005200              INC      R0          :UPDATE LINE POINTER
1695 011776 005302              DEC      R2          :4 LINES DONE?
1696 012000 001220              BNE      1$          :BR IF NO
1697 012002 000207              RTS      PC          :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,TSTNO		
1716	012012	012737	012512	001216		MOV	#TST6,NEXT		
1717	012020	012700	000000			MOV	#0,R0		:PLACE LINE NUMBER INTO R0
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		100\$:	MOV	#4,R0		:PLACE LINE NUMBER INTO R0
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		101\$:	MOV	#8,R0		: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		102\$:	MOV	#12,R0		: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			103\$:	SCOPE			:SCOPE THIS TEST.
1734	012122				105\$:				:TEST ENTRANCE.
1735	012122	012737	012156	001220		MOV	#15,LOCK		:SET FOR "LOCK ON LINE".
1736	012130	104413				RAMCLR			:CLEAR ALL SEC REGISTERS
1737	012132	105037	023611			CLRB	TXTAB+31		:CLEAT CNTRL BYTE (TX)
1738	012136	112737	000010	030211		MOVB	#BIT3,RXTAB+31		:SET "INC/BCC" IN RX CNTRL B TE
1739	012144	112737	000031	022560		MOVB	#31,TXBAP		:SET TX DATA CHAR
1740	012152	012702	000004			MOV	#4,R2		:4 LINE GROUP
1741	012156				1\$:				
1742	012156	104412			64\$:	MSTCLR			:INIT DV11
1743	012160	010077	167206			MOV	R0,ADVSR5		: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		
1747	012176	000000				0			:RXBCC, LINE PROTOCOL
1748	012200	000000				0			:START BCC AT 0.
1749	012202	005277	167154			INC	ADVSCR		:POLONOMINAL SELECT
1750	012206	105777	167150		65\$:	TSTB	ADVSCR		:SET MICRO CPU GO
1751	012212	100375				BPL	65\$		:WAIT FOR
1752	012214	112777	000007	167152		MOVB	#7,ADVSR5H		:BIT 7 OF DVSCR=1
1753	012222	017704	167150			MOV	ADVSR5,R4		:SEL RX BCC REG
1754	012226	005037	022116			CLR	CALBCC		:READ BCC
1755	012232	012737	000200	022112		MOV	#LRCB,XPOLY		:SET SOFTWARE BCC=0
1756	012240	004537	021740			JSR	R5,SIMBCC		:SET SOFTWARE POLONOMINAL
1757	012244	000010				8.			:GO GET SOFTWARE BCC
1758	012246	000031				31			:SHIFTS
1759	012250	000000				0			:DATA
1760	012252	013705	022115			MOV	CALBCC,R5		:PREVIOUS BCC
1761	012256	020504				CMP	R5,R4		:GET SOFTWARE BCC
1762	012260	001401				BEQ	.+4		:SOFT=HARD?
1763	012262	104004				HLT	4		
1764	012264	104412			66\$:	MSTCLR			:RECEIVER BCC INCORRECT!
1765	012266	010077	167100			MOV	R0,ADVSR5		:INIT DV11
1766	012272	004737	022224			JSR	PC,DV110N		:LOAD LINE NO.
1767	012276	004537	022120			PERFORM	SETREG		:GOSUB DV110N
1768	012302	007	012			.BYTE	007,012		
1769	012304	000000				0			:RXBCC, LINE PROTOCOL
1770	012306	000010				BIT3			:START BCC AT 0.
									:POLONOMINAL SELECT

1771	012310	005277	167046		INC	2DVSCR	:SET MICRO CPU GO
1772	012314	105777	167042	67\$:	TSTB	2DVSCR	:WAIT FOR
1773	012320	100375			BPL	67\$	:BIT 7 OF DVSCR=1
1774	012322	112777	000007	167044	MOVB	#7,2DVSRSH	:SEL RX BCC REG
1775	012330	017704	167042		MOV	2DVSR,R4	:READ BCC
1776	012334	005037	022116		CLR	CALBCC	:SET SOFTWARE BCC=0
1777	012340	012737	120001	022112	MOV	#CRC16,XPOLY	:SET SOFTWARE POLONOMINAL
1778	012346	004537	021740		JSR	R5,SIMBCC	:GO GET SOFTWARE BCC
1779	012352	000010			B.		:SHIFTS
1780	012354	000031			31		:DATA
1781	012356	000000			0		:PREVIOUS BCC
1782	012360	013705	022116		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	166772		MOV	R0,2DVSR	:LOAD LINE NO.
1788	012400	004737	022224		JSR	PC,DV11ON	:GOSUB DV11ON
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		:POLONOMINAL SELECT
1793	012416	005277	166740		INC	2DVSCR	:SET MICRO CPU GO
1794	012422	105777	166734	69\$:	TSTB	2DVSCR	:WAIT FOR
1795	012426	100375			BPL	69\$	:BIT 7 OF DVSCR=1
1796	012430	112777	000007	166736	MOVB	#7,2DVSRSH	:SEL RX BCC REG
1797	012436	017704	166734		MOV	2DVSR,R4	:READ BCC
1798	012442	005037	022116		CLR	CALBCC	:SET SOFTWARE BCC=0
1799	012446	012737	102010	022112	MOV	#CRC.CCITT,XPOLY	:SET SOFTWARE POLONOMINAL
1800	012454	004537	021740		JSR	R5,SIMBCC	:GO GET SOFTWARE BCC
1801	012460	000010			B.		:SHIFTS
1802	012462	000031			31		:DATA
1803	012464	000000			0		:PREVIOUS BCC
1804	012466	013705	022116		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			SCOP1		:LOCK ON SELECTED LINE
1809	012502	005200			INC	R0	:UPDATE LINE NO. POINTER
1810	012504	005302			DEC	R2	:ALL LINES DONE?
1811	012506	001223			BNE	1\$	:BR IF NO
1812	012510	000207			RTS	PC	:EXIT FOR NEXT GROUP

```

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

```

; TEST 6

1825	012512	012737	000006	001226	↑ST6:	MOV	#6,TSTNO
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	004737	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	100402			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 THIS TEST.
1844	012630				105\$:		TEST ENTRANCE.
1845	012630	012737	012662	001220	MOV	#15, 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 FOR 4 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	2DVSCR	SET MICRO CPU GO
1859	012710	105777	166446		TSTB	2DVSCR	WAIT FOR
1860	012714	100375			BPL	.-4	DVSCR07=1
1861	012716	112777	000015	166450	MOV	#15, 2DVSRSH	SEL RX MODE REGISTER
1862	012724	017704	166446		MOV	2DVSRSH, 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			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      ;*TEST OF THE "SEND DLE NEXT" FUNCTION
1884      ;*THE "TRANSMITTER DLE REGISTER" IS LOADED
1885      ;*WITH CHAR "025". THE RECEIVER IS SET TO RECEIVE
1886      ;*ONE CHAR (THE DLE) SO RX BA S/B=25
1887      ;*THE TRANSMITTER DATA CHAR IS "031".
1888      ;*****
1889
1890      ; TEST 7
1891      ;-----
1892      012772 012737 000007 001226 1ST7: MOV      #7,TSTNO
1893      013000 012737 013244 001216      MOV      #TST10,NEXT
1894      013006 012700 000000      MOV      #0.,RO
1895      013012 013737 001416 001236      MOV      L00.03,STAT
1896      013020 100402      BMI      100$
1897      013022 004737 013110      JSR      PC,105$
1898      013026 012700 000004      MOV      #4.,RO
1899      013032 013737 001420 001236 100$: MOV      L04.07,STAT
1900      013040 100402      BMI      101$
1901      013042 004737 013110      JSR      PC,105$
1902      013046 012700 000010      MOV      #8.,RO
1903      013052 013737 001422 001236 101$: MOV      L08.11,STAT
1904      013060 100402      BMI      102$
1905      013062 004737 013110      JSR      PC,105$
1906      013066 012700 000014      MOV      #12.,RO
1907      013072 013737 001424 001236 102$: MOV      L12.15,STAT
1908      013100 100402      BMI      103$
1909      013102 004737 013110      JSR      PC,105$
1910      013106 104400      103$: SCOPE
1911      013110      105$:
1912      013110 012737 013144 001220      MOV      #1$,LOCK
1913      013116 104413      RAMCLR
1914      013120 112737 000002 023611      MOV      #BIT1,TXTAB+31
1915      013126 112737 000031 022560      MOV      #31,TXBAP
1916      013134 105037 030211      CLR      RXTAB+31
1917      013140 012702 000004      MOV      #4,R2
1918      013144 004737 022224      1$: JSR      PC,DV11ON
1919      013150 004537 022120      PERFORM SETREG
1920      013154 012      .BYTE 012,012
1921      013156 012400      .BYTE 25*400
1922      013160 012400      .BYTE 25*400
1923      013162 005037 027560      CLR      RXBA
1924      013166 005277 166170      INC      @DVSCR
1925      013172 105777 166164      TSTB     @DVSCR
1926      013176 100375      BPL      -4
1927      013200 013704 027560      MOV      RXBA,R4
1928      013204 012705 000025      MOV      #25,R5
1929      013210 020504      CMP      R5,R4
1930      013212 001401      BEQ      2$
1931      013214 104003      HLT
1932      013216 104412      2$: MSTCLR
1933      013220 112777 000012 166146      MOV      #12,@DVSRSH
1934      013226 005077 166144      CLR      @DV5RA
1935      013232 104401      SCOP1
1936      013234 005200      INC      RO
1937      013236 005302      DEC      R2
1938      013240 001341      BNE      1$

```

```

; PLACE LINE NUMBER INTO RO
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 1
; PLACE LINE NUMBER INTO RJ
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 2
; LOAD LINE NUMBER
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TEST FOR LINE CARD 3
; LOAD LINE NO.
; LOAD LINE CARD STATUS
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TESTS FOR LINE CARD 4
; SCOPE THIS TEST.
; TEST ENTRANCE.
; SET IF SW09=1
; CLEAR ALL SEC REGISTERS
; SET "SND/DLE" IN CNTRL BYTE
; SET TX DATA CHAR
; ZERO RX CNTRL BYTE
; SET FOR 4 LINE GROUP
; SET ROUTINE THING
; LINE PROTOCOL REG
; PUT 25
; IN HIGH BYTE
; ZERO RX BUFFER
; SET MICRO CPU GO
; WAIT FOR
; DVSCR07=1
; GET DATA
; LOAD DLE INTO EXPECTED
;
; 25 (DLE) NOT 1ST IN RX BUFFER
; INIT DV11
; SEL LINE PROTOCOL
; ZERO IT.
; LOCK ON SELECTED LINE?
; UPDATE LINE POINTER
; 4 LINE GROUP DONE?
; BR IF NO

```

1939 013242 000207

RTS PC

;EXIT FOR NEXT GROUP OF LINES

1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950

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

; TEST 10

1951					ST10:	MOV	#10, TSTNO	
1952	013244	012737	000010	001226		MOV	#TST11, NEXT	
1953	013252	012737	013554	001216		MOV	#0, R0	
1954	013260	012700	000000			MOV	LO0.03, STAT	; PLACE LINE NUMBER INTO FJ
1955	013264	013737	001416	001236		MOV	100\$	; 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, R0	; PLACE LINE NUMBER INTO R0
1959	013304	013737	001420	001236		MOV	LO4.07, 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, R0	; LOAD LINE NUMBER
1963	013324	013737	001422	001236		MOV	LO8.11, 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, R0	; LOAD LINE NO.
1967	013344	013737	001424	001236		MOV	LI2.15, STAT	; LOAD LINE CARD STATUS
1968	013352	100402				BMI	103\$	; BR IF LINE CARD NOT TO BE TESTED
1969	013354	004737	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	#15, 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		MOV	#BIT7+BIT6+BITS, TXTAB+15	; SET RX AND TX NEXT MODE=7
1976	013404	113737	023575	030175		MOV	TXTAB+15, RXTAB+15	
1977	013412	004737	022224		1\$:	JSR	PC, DV110N	; SET UP MINOR DETAILS
1978	013416	112777	000012	165750		MOV	#12, DVVSRSH	; GET LINE PROTOCOL REGISTER
1979	013424	052777	000140	165744		BIS	#BIT6+BITS, DVVSR	; SET TX AND RX DDCMP MODE
1980	013432	112737	000015	022560		MOV	#15, TXBAP	; LOAD DATA CHAR
1981	013440	005277	165716			INC	DVSCR	; SET MICRO CPU GO
1982	013444	105777	165712			TSTB	DVSCR	; WAIT FOR
1983	013450	100375				BPL	-4	; DVSCR07=1
1984	013452	005005				CLR	R5	; EXPECTED=0
1985	013454	112777	000014	165712		MOV	#14, DVVSRSH	; GET TX MODE REG
1986	013462	017704	165710			MOV	DVVSRA, R4	; READ MODE REG
1987	013466	001401				BEQ	+4	; S/B=0
1988	013470	104001				HLT	1	; TX MODE REG S/B=0
1989	013472	105277	165676			INCB	DVVSRA, R4	; GET RX MODE REG
1990	013476	017704	165674			MOV	DVVSRA, R4	; READ RX MODE
1991	013502	001401				BEQ	3\$	
1992	013504	104001				HLT	1	; RX MODE REG S/B=0
1993	013506	112777	000006	165660	3\$:	MOV	#6, DVVSRSH	; TX BCC REG.
1994	013514	017704	165656			MOV	DVVSRA, R4	; READ TXBCC REG.

1995	013520	001001		BNE	4\$		:S/B NOT=0
1996	013522	104000		HLT	0		:NOTHING IN BCC!! (TX)
1997	013524	105277	165644	4\$: INCB	0DVSRSH		:GET RX BCC REG
1998	013530	017704	165642	MOV	0DV5RA,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		SCOPI			: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

```

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

```

2019							: TEST 11
2020							:-----
2021							:ST11: MOV #11,TSTNO
2022	013554	012737	000011	001226			:MOV #12,NEXT
2023	013562	012737	014166	001216			:MOV #0,R0
2024	013570	012700	000000				:PLACE LINE NUMBER INTO R0
2025	013574	113737	001406	001244			:PLACE "MASK" FOR CHARS INTO MASKX
2026	013602	013737	001416	001236			:LOAD LINE CARD STATUS INTO STAT
2027	013610	100402					:BR IF LINE CARD NOT TO BE TESTED
2028	013612	004737	013722				:GO DO THE TEST FOR LINE CARD 1
2029	013616	012700	000004		100\$:		:PLACE LINE NUMBER INTO R0
2030	013622	113737	001407	001244			:GET MASK
2031	013630	013737	001420	001236			:LOAD LINE CARD STATUS INTO STAT
2032	013636	100402					:BR IF LINE CARD NOT TO BE TESTED
2033	013640	004737	013722				:GO DO THE TEST FOR LINE CARD 2
2034	013644	012700	000010		101\$:		:LOAD LINE NUMBER
2035	013650	113737	001410	001244			:GET MASK
2036	013656	013737	001422	001236			:LOAD LINE CARD STATUS INTO STAT
2037	013664	100402					:BR IF LINE CARD NOT TO BE TESTED
2038	013666	004737	013722				:DO THE TEST FOR LINE CARD 3
2039	013672	012700	000014		102\$:		:LOAD LINE NO.
2040	013676	113737	001411	001244			:GET MASK
2041	013704	013737	001424	001236			:LOAD LINE CARD STATUS
2042	013712	100402					:BR IF LINE CARD NOT TO BE TESTED
2043	013714	004737	013722				:DO THE TESTS FOR LINE CARD 4
2044	013720	104400			103\$:		:SCOPE THIS TEST.
2045	013722				105\$:		:TEST ENTRANCE.
2046	013722	032737	004000	001236			:IS THIS AN ASYNC LINE CARD?
2047	013730	001401					:BR IF NOT ASYNC LINE CARD.
2048	013732	000207					:EXIT TEST. (ASYNC LINE CARD NOT TESTED)
2049	013734	012737	014004	001220			:SET RETURN IF SW09=1
2050	013742	104413					: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\$:	MOV#	STAT,(R5)+	:8. SYNC
2055	013764	005304			DEC	R4	:CHARS
2056	013766	001374			BNE	1\$	
2057	013770	112725	000005		MOV#	#5,(R5)+	:LOAD "NON-SYNC" CHAR
2058	013774	113725	001236		MOV#	STAT,(R5)+	:SYNC
2059	014000	113725	001236		MOV#	STAT,(R5)+	:SYNC
2060	014004	004737	022224	2\$:	JSR	PC,DV11ON	: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	MOV#	#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	:DVSCR07=1
2072	014060	012705	000005		MOV	#5,R5	:1ST DATA S/B=15
2073	014064	113704	027560		MOV#	RXBA,R4	:GET DATA
2074	014070	042704	177400		BIC	#1C<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\$:	MOV#	STAT,R5	:LOAD SYNC INTO EXPECTED
2079	014106	042705	177400		BIC	#1C<377>,R5	:STRIP HIGH BYTE
2080	014112	143705	001244		BICB	MASKX,R5	:CLEAR BITS/PER/CHAR MASK.
2081	014116	113704	027561		MOV#	RXBA+1,R4	:GET 2ND CHAR
2082	014122	042704	177400		BIC	#1C<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* SYNCs ARE TO BE STRIPPED
2086	014134	113704	027562	4\$:	MOV#	RXBA+2,R4	:GET 3RD CHAR
2087	014140	042704	177400		BIC	#1C<377>,R4	:STRIP HIGH BYTE
2088	014144	020504			CMP	R5,R4	:OK?
2089	014146	001401			BEQ	5\$	
2090	014150	104001			HLT	1	:JUST *LEADING* SYNCs 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	005302			DEC	R2	:4 LINE GROUP DONE?
2095	014162	001310			BNE	2\$	:BR IF NO
2096	014164	000207			RTS	PC	:EXIT FOR NEXT GROUP

```

***** 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
: *INTERRUPT FLAG.
: *THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.

```

2097
2098
2099
2100
2101
2102
2103
2104
2105
2106

```

2107
2108
2109
2110
2111 014166 012737 000012 001226
2112 014174 012737 014650 001216
2113 014202 012700 000000
2114 014206 013737 001416 001236
2115 014214 100402
2116 014216 004737 014304
2117 014222 012700 000004 100$:
2118 014226 013737 001420 001236
2119 014234 100402
2120 014236 004737 014304
2121 014242 012700 000010 101$:
2122 014246 013737 001422 001236
2123 014254 100402
2124 014256 004737 014304
2125 014262 012700 000014 102$:
2126 014266 013737 001424 001236
2127 014274 100402
2128 014276 004737 014304
2129 014302 104400 103$:
2130 014304 104400 105$:
2131 014304 012737 014362 001220
2132 014312 104413
2133 014314 005001
2134 014316 012702 000004
2135 014322 005005
2136 014324 012704 022560
2137 014330 110524 1$:
2138 014332 005205
2139 014334 022705 000007
2140 014340 001373
2141 014342 012705 000006
2142 014346 012704 030160
2143 014352 112724 000001 2$:
2144 014356 005305
2145 014360 001374
2146 014362 004737 022224 66$:
2147 014366 004537 022120
2148 014372 001 005
2149 014374 177770
2150 014376 177772
2151 014400 112777 000012 164766
2152 014406 052777 000100 164762
2153 014414 012737 000340 177776
2154 014422 012777 014472 164722
2155 014430 012777 000340 164716
2156 014436 052777 000101 164716
2157
2158 014444 005003
2159 014446 005037 014646 3$:
2160 014452 005037 177776
2161 014456 104414
2162 014460 005237 014646

```

;*****

; TEST 12

```

ST12: MOV #12,TSTNO
MOV #TST13,NEXT
MOV #0,R0
MOV L00.03,STAT
BMI 100$
JSR PC,105$
100$: MOV #4,R0
MOV L04.07,STAT
BMI 101$
JSR PC,105$
101$: MOV #8,R0
MOV L08.11,STAT
BMI 102$
JSR PC,105$
102$: MOV #12,R0
MOV L12.15,STAT
BMI 103$
JSR PC,105$
103$: SCOPE
105$:
MOV #66$,LOCK
RAMCLR
CLR R1
MOV #4,R2
CLR R5
MOV #TXBAP,R4
1$: MOV R5,(R4)+
INC R5
CMP #7,R5
BNE 1$
MOV #6,R5
MOV #RXTAB,R4
2$: MOV R5,(R4)+
DEC R5
BNE 2$
66$: JSR PC,DV11ON
PERFORM SETREG
.BYTE 001,005
-8.
-6.
MOVB #12,ADVSRSH
BIS #BIT6,ADVSRSH
MOV #340,PS
MOV #67$,ADVREVC
MOV #340,ADVRLVL
BIS #BIT6+BIT0,ADVSCA
SET RX IE AND UCPU GO.
DATA IMAGE
STALL COUNTER
ENABLE INTERRUPTS
WASTE TIME
UPDATE STALL

```

```

; PLACE LINE NUMBER INTO R0
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 1
; PLACE LINE NUMBER INTO R0
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 2
; LOAD LINE NUMBER
; LOAD LINE CARD STATUS IF 0 STAT
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TEST FOR LINE CARD 3
; LOAD LINE NO.
; LOAD LINE CARD STATUS
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TESTS FOR LINE CARD 4
; SCOPE THIS TEST.
; TEST ENTRANCE.

```

; CLEAR ALL SEC REGISTERS

```

; SET FOR 4 LINE GROUP
; LOAD
; TX DATA

```

```

; FILL
; THE
; RX CNTRL TABLE
; WITH SPECIAL
; CHAR BITS
; SET UP DV11

```

; TX PRINCIPLE BC, RX BC

```

; LINE PROTOCOL
; TX DDCMP
; LOCK OUT INTERRUPTS
; SET RX INTER VECTOR.
; SET PRIO. LEVEL TO 7

```

```

; SET RX IE AND UCPU GO.
; DATA IMAGE
; STALL COUNTER
; ENABLE INTERRUPTS
; WASTE TIME
; 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	010005	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	2DVHIC,R4	;READ FOUND RESULT
2170	014504	020504		CMP	R5,R4	
2171	014506	001401		BEQ	4\$	
2172	014510	104001		HLT	1	;DVRIC WRONG!
2173	014512	052777	000400 164642 4\$:	BIS	#BIT8,2DVSCR	;SET "RECEIVER INT RESP COMP"
2174	014520	005203		INC	R3	;UPDATE DATA IMAGE
2175	014522	022703	000006	CMP	#6,R3	;ALL DONE?
2176	014526	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	#BIT6,2DVSCR	;NO MORE INTERRUPTS.
2180	014544	012716	014552	MOV	#68\$, (SP)	;SET RETURN
2181	014550	000002		RTI		;CONT.
2182	014552	105777	164604 68\$:	TSTB	2DVSCR	;WAIT FOR
2183	014556	100375		BPL	-4	;DVSC07=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	2DVRIC,R4	;READ RESULTS
2188	014574	020504		CMP	R5,R4	
2189	014576	001401		BEQ	5\$	
2190	014600	104001		HLT	1	;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		MOVB	(R1)+,R4	;GET RX DATA (INCORE)
2195	014614	020504		CMP	R5,R4	
2196	014616	001401		BEQ	7\$	
2197	014620	104001		HLT	1	;RECEIVER PLACED DATA IN CORE WRONG
2198	014622	005205		INC	R5	;UPDATE DATA IMAGE
2199	014624	022705	000006 7\$:	CMP	#6,R5	;ALL DONE?
2200	014630	001370		BNE	6\$	;BR IF NO
2201	014632	104412		MSTCLR		;INIT DV11
2202	014634	104401		SCOPI		;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	66\$	;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 "8" 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 1ST13: MOV      #13,TSTNO
2230      014656 012737 015442 001216      MOV      #TST14,NEXT
2231      014664 012700 000000      MOV      #0,R0
2232      014670 013737 001416 001236      MOV      L00.03,STAT
2233      014676 100402      BMI      100$
2234      014700 004737 014766      JSR      PC,105$
2235      014704 012700 000004      100$: MOV      #4,R0
2236      014710 013737 001420 001236      MOV      L04.07,STAT
2237      014716 100402      BMI      101$
2238      014720 004737 014766      JSR      PC,105$
2239      014724 012700 000010      101$: MOV      #8,R0
2240      014730 013737 001422 001236      MOV      L08.11,STAT
2241      014736 100402      BMI      102$
2242      014740 004737 014766      JSR      PC,105$
2243      014744 012700 000014      102$: MOV      #12,R0
2244      014750 013737 001424 001236      MOV      L12.15,STAT
2245      014756 100402      BMI      103$
2246      014760 004737 014766      JSR      PC,105$
2247      014764 104400      103$: SCOPE
2248      014766      105$:
2249      014766 012737 015114 001220      MOV      #65$,LOCK
2250      014774 032737 001400 001236      BIT      #BIT9+BIT8,STAT
2251      015002 001401      BEQ      +4
2252      015004 000207      RTS      PC
2253      015006 104413      RAMCLR
2254      015010 012702 000004      MOV      #4,R2
2255      015014 012704 000012      MOV      #10,R4
2256      015020 012705 023560      MOV      #TXTAB,R5
2257      015024 112725 000010      1$: MOVVB   #BIT3,(R5)+
2258      015030 005304      DEC      R4
2259      015032 001374      BNE      1$
2260      015034 012705 023560      MOV      #TXTAB,R5
2261      015040 013704 001236      MOV      STAT,R4
2262      015044 042704 177400      BIC      #C<377>,R4
2263      015050 060405      ADD      R4,R5
2264      015052 105015      CLRB    (R5)
2265      015054 012705 022560      MOV      #TXBAP,R5
2266      015060 005004      CLR      R4
2267      015062 110425      2$: MOVVB   R4,(R5)+
2268      015064 005204      INC      R4
2269      015066 022704 000013      CMP      #11,R4
2270      015072 001373      BNE      2$
2271      015074 012705 030160      MOV      #RXTAB,R5
2272      015100 012704 000012      MOV      #10,R4
2273      015104 112725 000010      3$: MOVVB   #BIT3,(R5)+
2274      015110 005304      DEC      R4

```

```

; PLACE LINE NUMBER INTO R0
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 1
; PLACE LINE NUMBER INTO R0
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 2
; LOAD LINE NUMBER
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TEST FOR LINE CARD 3
; LOAD LINE NO.
; LOAD LINE CARD STATUS
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TESTS FOR LINE CARD 4
; SCOPE THIS TEST.
; TEST ENTRANCE.
; SET RETURN IF SW09=1
; "8 BITS/PER/CHAR ?"
; BR IF YES
; EXIT TEST FOR THIS LINE CARD!
; CLEAR ALL SECONDARY REGISTERS
; SET FOR 4 LINE GROUP
; LOAD 10 BYTES
; WITH
; INC/BCC
;
; CLEAR
; SYNC
; CONTROL
; BYTE
;
; LOAD
; DATA
; INTO
; TRANSMITTER BUFFER
;
; LOAD
; 10
; RECEIVER
; CONTROL BYTES

```

2275	015112	001374			BNE	3\$	:WITH "INC/BCC"
2276	015114	010077	164252		MOV	RO,ADVSR5	:LOAD LINE NO.
2277	015120	032737	004000	001236	BIT	#ASYNC,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		PERFORM	SETREG	
2285	015150	000	001		.BYTE	000,001	:TX PRINCIPLE BA. BC
2286	015152	022556			SYNC		
2287	015154	077764			<-12.>-BIT15		:MARKED BC!
2288	015156	004537	022120		PERFORM	SETREG	
2289	015162	004	005		.BYTE	004,005	:RX BA, BC
2290	015164	027560			RXBA		
2291	015166	077766			<-10.>-BIT15		:MARKED BC!
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	#ASYNC,STAT	:IS THIS ASYNC LINE CARD?
2305	015234	001412			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		PERFORM	LOAD.MODE	:LOAD
2315	015266	034000			BIT13+BIT12+BIT11		:MODE AND RECV ENABLE
2316	015270	004537	021706		PERFORM	SETSYNC	:GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2317	015274	005277	164062		INC	ADVSCR	:SET MICRO CPU GO
2318	015300	105777	164056		TSTB	ADVSCR	:WAIT FOR
2319	015304	100375			BPL	-4	:DVSCRO7=1
2320	015306	017704	164054		MOV	ADVRC,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	MOV	#14,ADVSR5H	:GET TX MODE REGISTER
2328	015336	017704	164034		MOV	ADVSR4,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	55		
2332	015352	104001			HLT	1		: NEXT MODE INCORRECT/ S/B=7
2333	015354	105277	164014	55:	INCB	2DVSRSH		: SEL RX MODE REG
2334	015360	017704	164012		MOV	2DVSR, R4		: READ
2335	015364	020504			CMP	R5, R4		
2336	015366	001401			BEQ	65		
2337	015370	104001			HLT	1		: RX MODE REGISTER INCORRECT. S/B=7
2338	015372	005005		65:	CLR	R5		: SET EXPECTED=0
2339	015374	112777	000006	163772	MOVB	#6, 2DVSRSH		: SEL TX BCC REG
2340	015402	017704	163770		MOV	2DVSR, R4		: READ
2341	015406	001401			BEQ	75		: BR IF=0
2342	015410	104001			HLT	1		: IF BCC WAS SENT; BCC S/B=0
2343	015412	105277	163756	75:	INCB	2DVSRSH		: SEL RX BCC REG
2344	015416	017704	163754		MOV	2DVSR, R4		: READ IT
2345	015422	001401			BEQ	85		
2346	015424	104001			HLT	1		: IF RX RECVD GOOD BCC; LJC S/B=C
2347	015426	104413		85:	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	655		: BR IF NO
2352	015440	000207			RTS	PC		: EXIT FOR NEXT 4 LINE GROUP

```

*****TEST 14*****
*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 IN THE TRANSMITTER MODE REG.
*ALSO VERIFY THAT BIT10=1 IN LINE STATE MAKES
*RECEIVER "EXPECT THE BCC"
*AND THAT BIT10 IN LINE PROGRESS TELL TX TO SEND BCC.
*THIS TEST USES LRCB FOR THE POLYNOMIAL.
*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
*****

```

2370								
2371								: TEST 14
2372	015442	012737	000014	001226	TEST14:	MOV	#14, TSTNO	
2373	015450	012737	016222	001216		MOV	#TST15, NEXT	
2374	015456	012700	000000			MOV	#0, R0	
2375	015462	013737	001416	001236		MOV	L00.03, STAT	: PLACE LINE NUMBER INTO R0
2376	015470	100402				BMI	1005	: LOAD LINE CARD STATUS INTO STAT
2377	015472	004737	015560			JSR	PC, 1055	: BR IF LINE CARD NOT TO BE TESTED
2378	015476	012700	000004		1005:	MOV	#4, R0	: GO DO THE TEST FOR LINE CARD 1
2379	015502	013737	001420	001236		MOV	L04.07, STAT	: PLACE LINE NUMBER INTO R0
2380	015510	100402				BMI	1015	: LOAD LINE CARD STATUS INTO STAT
2381	015512	004737	015560			JSR	PC, 1055	: BR IF LINE CARD NOT TO BE TESTED
2382	015516	012700	000010		1015:	MOV	#8, R0	: GO DO THE TEST FOR LINE CARD 2
2383	015522	013737	001422	001236		MOV	L08.11, STAT	: LOAD LINE NUMBER
2384	015530	100402				BMI	1025	: LOAD LINE CARD STATUS INTO STAT
2385	015532	004737	015560			JSR	PC, 1055	: BR IF LINE CARD NOT TO BE TESTED
2386	015536	012700	000014		1025:	MOV	#12, R0	: DO THE TEST FOR LINE CARD 3

: 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			SCOPE		:SCOPE THIS TEST.
2391	015560				103\$:		:TEST ENTRANCE.
2392	015560	012737	015674	001220	105\$:		:SET RETURN IF SW09=1
2393	015566	104413			MOV	#65\$,LOCK	:CLEAR ALL SECONDARY REGISTERS
2394	015570	012702	000004		RAMCLR		:SET FOR 4 LINE GROUP
2395	015574	012704	000012		MOV	#4,R2	:LOAD 10 BYTES
2396	015600	012705	023560		MOV	#10,R4	:WITH
2397	015604	112725	000010		MOV	#TXTAB,R5	:INC/BCC
2398	015610	005304			1\$:	#BIT3,(R5)+	
2399	015612	001374			DEC	R4	
2400	015614	012705	023560		BNE	1\$	
2401	015620	013704	001236		MOV	#TXTAB,R5	:CLEAR
2402	015624	042704	177400		MOV	STAT,R4	:SYNC
2403	015630	060405			BIC	#C<377>,R4	:CONTROL
2404	015632	105015			ADD	R4,R5	:BYTE
2405	015634	012705	022560		CLRB	(R5)	
2406	015640	005004			MOV	#TXBAP,R5	:LOAD
2407	015642	110425			CLR	R4	:DATA
2408	015644	005204			2\$:	R4,(R5)+	:INTO
2409	015646	022704	000013		INC	R4	:TRANSMITTER BUFFER
2410	015652	001373			CMP	#11.,R4	
2411	015654	012705	030160		BNE	2\$	
2412	015660	012704	000012		MOV	#RXTAB,R5	:LOAD
2413	015664	112725	000010		MOV	#10,R4	:10
2414	015670	005304			3\$:	MOV#BIT3,(R5)+	:RECEIVER
2415	015672	001374			DEC	R4	:CONTROL BYTES
2416	015674	010077	163472		BNE	3\$	:WITH "INC/BCC"
2417	015700	032737	004000	001236	65\$:	MOV RD,DOVSRS	:LOAD LINE NO.
2418	015706	001406			BIT	#ASYNC,STAT	:IS THIS AN ASYNC LINE CARD?
2419	015710	004537	022120		BEQ	80\$	:BR IF NOT ASYNC.
2420	015714	000	001		PERFORM	SETREG	:ADJUST FOR ASYNC LINE CARD
2421	015716	022560			.BYTE	000,001	:REGISTERS
2422	015720	077766			TXBAP		:LOAD FOR ASYNC
2423	015722	000405			<-10.>-BIT15		:LOAD FOR ASYNC
2424	015724	004537	022120		BR	81\$	:CONTINUE TEST
2425	015730	000	001		80\$:	PERFORM SETREG	
2426	015732	022556			.BYTE	000,001	:TX PRINCIPLE BA, BC
2427	015734	077764			SYNC		
2428	015736	004537	022120		<-12.>-BIT15		:MARKED BC!
2429	015742	004	005		81\$:	PERFORM SETREG	
2430	015744	027560			.BYTE	004,005	:RX BA, BC
2431	015746	077766			RXBA		
2432	015750	004537	022120		<-10.>-BIT15		:MARKED B.!
2433	015754	010	011		PERFORM	SETREG	
2434	015756	023560			.BYTE	010,011	:TX TABLE, RX TABLE
2435	015760	030160			TXTAB		
2436	015762	004537	022120		RXTAB		
2437	015766	012	013		PERFORM	SETREG	
2438	015770	000001			.BYTE	012,013	:LINE PROTOCOL, LINE STATE
2439	015772	162004			BIT0		:LRCC, IDLE MARK
2440	015774	004537	022120		BIT15+BIT14+BIT13+BIT10+BIT2		
2441	016000	016	017		PERFORM	SETREG	:MODE 7, TXGO
2442	016002	162000			.BYTE	016,017	:LINE PROGRESS REC. REC CNTR STORE
					BIT15+BIT14+BIT13+BIT10		:NEXT MODE=7

```

2443 016004 000000 0 ;ZERO
2444 016006 032737 004000 001236 BIT #ASYNC,STAT ;IS THIS ASYNC 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 ;8 BITS/PER/CHAR
2450 016032 004537 022164 PERFORM ,LOAD.MODE ;
2451 016036 072000 <BIT14+BIT13+BIT12>+BIT10 ;
2452 ;9600 BAUD.
2453 016040 000405 BR 61$
2454 016042 004537 022164 60$: PERFORM ,LOAD.MODE ;LOAD
2455 016046 034000 BIT13+BIT12+BIT11 ;MODE AND RECV ENABLE
2456 016050 004537 021706 PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2457 016054 005277 163302 61$: INC ,DVSCR ;SET MICRO CPU GO
2458 016060 105777 163276 TSTB ,DVSCR ;WAIT FOR
2459 016064 100375 BPL -4 ;DVSCR07=1
2460 016066 017704 163274 MOV ,DVRIC,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$: MOVB #14 ,DVSRSH ;GET TX MODE REGISTER
2468 016116 017704 163254 MOV ,DVSR,R4
2469 016122 012705 000007 MOV #BIT2+BIT1+BIT0,R5 ;WAS NEXT MODE PICKED UP?
2470 016126 020504 CMP R5,R4
2471 016130 001401 BEQ 5$
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$: MOVB #6 ,DVSRSH ;SEL TX BCC REG
2480 016162 017704 163210 MOV ,DVSR,R4 ;READ
2481 016166 001401 BEQ 7$ ;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 8$: RAMCLR ;CLEAR ALL SEC REG
2488 016210 104401 SCOPI ;LOCK ON CURRENT LINE?
2489 016212 005200 INC R0 ;UPDATE LINE POINTER
2490 016214 005302 DEC R2 ;4 LINE GROUP DONE?
2491 016216 001226 BNE 65$ ;BR IF NO
2492 016220 000207 RTS PC ;EXIT FOR NEXT 4 LINE GROUP

;***** TEST 15 *****
;*TEST OF RECIEVER AND TRANSMITTER MODE BITS.
;*TEST TO TRANSMIT AND RECEIVE
;*4 DIFFERENT CHAR FROM EACH

```

2499
 2500
 2501
 2502
 2503
 2504
 2505
 2506
 2507
 2508
 2509
 2510
 2511
 2512
 2513
 2514
 2515
 2516
 2517
 2518
 2519
 2520
 2521
 2522
 2523
 2524
 2525
 2526
 2527
 2528
 2529
 2530
 2531
 2532
 2533
 2534
 2535
 2536
 2537
 2538
 2539
 2540
 2541
 2542
 2543
 2544
 2545
 2546
 2547
 2548
 2549
 2550
 2551
 2552
 2553
 2554

```

:MODE. THE TX TABLE WILL BE
:FILLED WITH "SEND DLE" SO IF CHAR
:GOES TO WRONG TABLE RX WILL
:RECEIVE A DLE CHAR(31). THE RX
:FILLS TABLE WITH "INCLUDE IN BCC"
:SO THAT IF RECV GOES TO WRONG
:TABLE THE RX BCC REG WILL
:BE NON-ZERO!
:CHAR CURRENT MODE NEXT MODE
: 15 0 1
: 16 1 2
: 21 2 3
: 23 3 4
: 25 4 5
: 7 5 6
: 34 6 7
: 32 7 7
: 36 7 7

```

:THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.

:*****

: TEST 15 -----

```

:ST15: MOV #15,TSTNO
      MOV #ST16,NEXT
      MOV #0,R0
      MOV L00.03,STAT
      BMI 100$
      JSR PC,105$
      MOV #4,R0
      MOV L04.07,STAT
      BMI 101$
      JSR PC,105$
      MOV #8,R0
      MOV L08.11,STAT
      BMI 102$
      JSR PC,105$
      MOV #12,R0
      MOV L12.15,STAT
      BMI 103$
      JSR PC,105$
      SCOPE
      105$: MOV #12$,LOCK
      RAMCLR
      MOV #TXTAB,R5
      MOV #RXTAB,R4
      MOV #4000,R1
      1$: MOV #BIT1,(R5)+
      2$: MOV #BIT3,(R4)+
      DEC R1
      BNE 1$
      PERFORM ,SETSYNC
      11$: MOV #4,R2

```

```

: PLACE LINE NUMBER INTO R0
: LOAD LINE CARD STATUS INTO STAT
: BR IF LINE CARD NOT TO BE TESTED
: GO DO THE TEST FOR LINE CARD 1
: PLACE LINE NUMBER INTO R0
: LOAD LINE CARD STATUS INTO STAT
: BR IF LINE CARD NOT TO BE TESTED
: GO DO THE TEST FOR LINE CARD 2
: LOAD LINE NUMBER
: LOAD LINE CARD STATUS INTO STAT
: BR IF LINE CARD NOT TO BE TESTED
: DO THE TEST FOR LINE CARD 3
: LOAD LINE NO.
: LOAD LINE CARD STATUS
: BR IF LINE CARD NOT TO BE TESTED
: DO THE TESTS FOR LINE CARD 4
: SCOPE THIS TEST.
: TEST ENTRANCE.
: LOCK ON LINE RETURN
: CLEAR ALL SEC REGISTERS
: LOAD
: ALL CNTRL BYTES
: WITH "SND/DLE"
: WITH "INCL/BCC"
:
: GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
: SET FOR 4 LINE GROUP

```

2555	016410	113705	001236		MOV B	STAT, R5	: CLEAR
2556	016414	042705	177400		BIC	#1C(377), R5	: SYNC
2557	016420	012704	023560		MOV	#TXAB, R4	: ENTRY
2558	016424	060504			ADD	R5, R4	: IN
2559	016426	105014			CLRB	(R4)	: CONTROL TABLE
2560	016430	112737	000040	023575	MOV B	#BIT5, TXAB+15	
2561	016436	112737	000100	024176	MOV B	#BIT6, TXAB+BIT8+16	
2562	016444	112737	000140	024601	MOV B	#BIT6+BIT5, TXAB+BIT9+21	
2563	016452	112737	000200	025203	MOV B	#BIT7, TXAB+BIT9+BIT8+23	
2564	016460	112737	000240	025605	MOV B	#BIT7+BIT5, TXAB+BIT10+25	
2565	016466	112737	000300	026167	MOV B	#BIT7+BIT6, TXAB+BIT10+BIT8+7	
2566	016474	112737	000340	026614	MOV B	#BIT7+BIT6+BIT5, TXAB+BIT10+BIT9+34	
2567	016502	112737	000340	027212	MOV B	#BIT7+BIT6+BIT5, TXAB+BIT10+BIT9+BIT8+32	
2568	016510	112737	000340	027216	MOV B	#BIT7+BIT6+BIT5, TXAB+BIT10+BIT9+BIT8+36	
2569							
2570	016516	112737	000040	030175	MOV B	#BIT5, RXAB+15	
2571	016524	112737	000100	030576	MOV B	#BIT6, RXAB+BIT8+16	
2572	016532	112737	000140	031201	MOV B	#BIT6+BIT5, RXAB+BIT9+21	
2573	016540	112737	000200	031603	MOV B	#BIT7, RXAB+BIT9+BIT8+23	
2574	016546	112737	000240	032205	MOV B	#BIT7+BIT5, RXAB+BIT10+25	
2575	016554	112737	000300	032587	MOV B	#BIT7+BIT6, RXAB+BIT10+BIT8+7	
2576	016562	112737	000340	033214	MOV B	#BIT7+BIT6+BIT5, RXAB+BIT10+BIT9+34	
2577	016570	112737	000340	033612	MOV B	#BIT7+BIT6+BIT5, RXAB+BIT10+BIT9+BIT8+32	
2578	016576	112737	000340	033616	MOV B	#BIT7+BIT6+BIT5, RXAB+BIT10+BIT9+BIT8+36	
2579	016604	012705	027560		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)+	: O
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 B	#36, (R5)+	: BUFFER
2595	016652	010077	162514		MOV	RO, ADVSR5	: 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		PERFORM	SETREG	
2604	016706	000	001		.BYTE	000, 001	: PRINCIPLE BA, BC
2605	016710	022556			SYNC		
2606	016712	177765			-11.		
2607	016714	004537	022120		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	011		.BYTE 010,011		TX TABLE, RX TAB
2613	016734	023560			TXTAB		
2614	016736	030160			RXTAB		
2615	016740	004537	022120		PERFORM SETREG		
2616	016744	012	013		.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			BEQ 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		#8 BITS/PER/CHAR
2625	016776	004537	022164		PERFORM LOAD.MODE		
2626	017002	072000			<BIT14+BIT13+BIT12>+BIT10		
2627							#9600 BAUD.
2628	017004	000403			BR 61\$		
2629	017006	004537	022164		PERFORM LOAD.MODE		LOAD
2630	017012	034000			BIT13+BIT12+BIT11		MODE AND RX ENABLE
2631	017014	005277	162342		INC 3DVSCR		SET MICRO CPU GO
2632	017020	105777	162336		TSTB 3DVSCR		WAIT FOR
2633	017024	100375			BPL -4		DVSCRO7=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	001246	MOV #9, TEMP1		CHECK 9. CHAR
2637	017044	005005			CLR R5		
2638	017046	005004			CLR R4		
2639	017050	112105			MOVB (R1)+, R5		SET EXPECTED
2640	017052	112304			MOVB (R3)+, R4		SET FOUND
2641	017054	020504			CMP R5, R4		GOOD?
2642	017056	001401			BEQ 4\$		
2643	017060	104001			HLT 1		DATA COMPARE ERROR (IS IT IDLE)?
2644	017062	005337	001246		DEC TEMP1		ALL CHARS DONE?
2645	017066	001370			BNE 3\$		BR IF NO
2646	017070	005005			CLR R5		
2647	017072	112777	000007	162274	MOVB #7, 3DVSRSH		SEL RX BCC REG
2648	017100	017704	162272		MOV 3DVSR, R4		READ IT
2649	017104	001401			BEQ 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, 3DVSRSH		SEL TX MODE REG
2653	017122	017704	162250		MOV 3DVSR, 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		INCB 3DVSRSH		SEL RX MODE REG
2658	017140	017704	162232		MOV 3DVSR, 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			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

```
:EXIT FOR NEXT GROUP OF LINES
```

```

***** TEST 16 *****
*TEST OF RECEIVER AND TRANSMITTER MULTIPLE FUNCTIONS.
*TEST OF RECV BCC AND TRANS BCC.
*CHAR      RX FUNC.      TX FUNC.
* 0          INC/BCC      INC/BCC
* 1          INC/BCC/DSCARD INC/BCC
* 2          INC/BCC      INC/BCC/SND/DLE
* 3          INC/BCC      INC/BCC
* 4          NO FUNC      SND/DLE
* 5          INC/BCC/DSCARD INC/BCC
* 6          INC/BCC/EXP/BCC INC/BCC/SND/BCC
*          NEXT MODE =7      NEXT MODE =7
*
*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE JARDS.
*****

```

: TEST 16

2687	017166	012737	00C016	001226	†ST16:	MOV	#16,TSTNO		
2688	017174	012737	020144	001216		MOV	†ST17,NEXT		
2689	017202	012700	000000			MOV	#0.,RO	:	PLACE LINE NUMBER INTO RO
2690	017206	013737	001416	001236		MOV	L00.03,STAT	:	LOAD LINE CARD STATUS INTO STAT
2691	017214	100402				BMI	100\$	:	BR IF LINE CARD NOT TO BE TESTED
2692	017216	004737	017304			JSR	PC,105\$	:	GO DO THE TEST FOR LINE CARD 1
2693	017222	012700	000004		100\$:	MOV	#4.,RO	:	PLACE LINE NUMBER INTO RO
2694	017226	013737	001420	001236		MOV	L04.07,STAT	:	LOAD LINE CARD STATUS INTO STAT
2695	017234	100402				BMI	101\$	:	BR IF LINE CARD NOT TO BE TESTED
2696	017236	004737	017304			JSR	PC,105\$	:	GO DO THE TEST FOR LINE CARD 2
2697	017242	012700	000010		101\$:	MOV	#8.,RO	:	LOAD LINE NUMBER
2698	017246	013737	001422	001236		MOV	L08.11,STAT	:	LOAD LINE CARD STATUS INTO STAT
2699	017254	100402				BMI	102\$	:	BR IF LINE CARD NOT TO BE TESTED
2700	017256	004737	017304			JSR	PC,105\$	:	DO THE TEST FOR LINE CARD 3
2701	017262	012700	000014		102\$:	MOV	#12.,RO	:	LOAD LINE NO.
2702	017266	013737	001424	001236		MOV	L12.15,STAT	:	LOAD LINE CARD STATUS
2703	017274	100402				BMI	103\$	:	BR IF LINE CARD NOT TO BE TESTED
2704	017276	004737	017304			JSR	PC,105\$	:	DO THE TESTS FOR LINE CARD 4
2705	017302	104400			103\$:		SCOPE	:	SCOPE THIS TEST.
2706	017304				105\$:			:	TEST ENTRANCE.
2707	017304	012737	017504	001220		MOV	#3\$,LOCK	:	RETURN IF SW09=1
2708	017312	032737	001400	001236		BIT	#BIT9+BIT8,STAT	:	"8 BITS/PER/CHAR ?"
2709	017320	001401				BEQ	+4	:	BR IF YES
2710	017322	000207				RTS	PC	:	EXIT TEST FOR THIS LINE CARD!
2711	017324	104413				RAMCLR		:	CLEAR ALL SEC REGISTERS
2712	017326	012705	023560			MOV	#TXTAB,R5	:	CLEAR
2713	017332	012703	030160			MOV	#RXTAB,R3	:	TRANSMITTER
2714	017336	005004				CLR	R4	:	AND
2715	017340	005025			1\$:	CLR	(R5)+	:	RECEIVER
2716	017342	005023				CLR	(R3)+	:	CONTROL
2717	017344	105204				INCB	R4	:	TABLES
2718	017346	100374				BPL	1\$	:	
2719	017350	012705	000010			MOV	#BIT3,R5	:	INC/BCC IS IN R5
2720	017354	110537	023560			MOVB	R5,TXTAB	:	INC/BCC
2721	017360	110537	023561			MOVB	R5,TXTAB+1	:	INC/BCC
2722	017364	110537	023562			MOVB	R5,TXTAB+2	:	INC/BCC

M05

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 54

SEG 0064

CZDVDC.P11 02-FEB-78 13:55

DV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1975 DIGITAL EQUIP. CORP.

2723	017370	152737	000002	023562	BISB	#BIT1, TXTAB+2	: INC/BCC	SND/DLE
2724	017376	110537	023563		MOVb	R5, TXTAB+3	: INC/BCC	
2725	017402	112737	000002	023564	MOVb	#BIT1, TXTAB+4	: INC/BCC	
2726	017410	110537	023565		MOVb	R5, TXTAB+5	: INC/BCC	
2727	017414	110537	023566		MOVb	R5, TXTAB+6	: INC/BCC	
2728	017420	052737	000344	023566	BIS	#BIT7+BIT6+BIT5+BIT2, TXTAB+6	: INC/BCC	; INC/BCC SND/BCC MOD=7
2729	017426	110537	030160		MOVb	R5, RXTAB	: INC/BCC	
2730	017432	110537	030161		MOVb	R5, RXTAB+1	: INC/BCC	
2731	017436	152737	000020	030161	BISB	#BIT4, RXTAB+1	: INC/BCC	DISCARD
2732	017444	110537	030162		MOVb	R5, RXTAB+2	: INC/BCC	
2733	017450	110537	030163		MOVb	R5, RXTAB+3	: INC/BCC	
2734	017454	105037	030164		CLRB	RXTAB+4	: NO FUNC.	
2735	017460	110537	030165		MOVb	R5, RXTAB+5	: INC/BCC	
2736	017464	152737	000020	030165	BISB	#BIT4, RXTAB+5	: INC/BCC	DISCARD
2737	017472	112737	000354	030166	MOVb	#BIT7+BIT6+BIT5+BIT3+BIT2, RXTAB+6	: INC/BCC	; INC/BCC EXP/BCC MODE=7
2738	017500	012702	000004		MOV	#4 R2	: SET FOR 4 LINE GROUP	
2739	017504	005037	027560		CLR	RXBA	: ZERO	
2740	017510	005037	027562	3\$:	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, 20VSRS	: LOAD LINE NO.	
2744	017530	032737	004000	001236	BIT	#ASYNC, STAT	: IS THIS AN ASYNC LINE CARD?	
2745	017536	001406			BEQ	60\$	: 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			-7.		: 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	010031			<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\$:	MOVB	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	2DVSCR	SET MICRO CODE GO
2787	017716	105777	161440		TSTB	2DVSCR	WAIT FOR
2788	017722	100375			BPL	-4	DVSCR07=1
2789	017724	012701	027560		MOV	#RXBA,R1	GET RX POINTER
2790	017730	012703	020134		MOV	#SOS,R3	GET DATA EXPECTED POINTER
2791	017734	012737	000007	001252	MOV	#7,TEMP3	CHECK 7 CHARS
2792	017742	112104		6\$:	MOVB	(R1)+,R4	GET RECEIVED CHAR
2793	017744	112305			MOVB	(R3)+,R5	GET EXPECTED CHAR
2794	017746	020504			CMP	R5,R4	OK?
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	MOVB	#14,2DVSRSH	GET TX MODE REG.
2800	017770	017704	161402		MOV	2DVSRSH,R4	
2801	017774	042704	177770		BIC	#1C<BIT2+BIT1+BIT0>,R4	CLEAR JUNK
2802	020000	012705	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	2DVSRSH	RX MODE REG
2807	020016	017704	161354		MOV	2DVSRSH,R4	
2808	020022	042704	177770		BIC	#1C<BIT2+BIT1+BIT0>,R4	
2809	020026	020504			CMP	R5,R4	
2810	020030	001401			BEQ	9\$	
2811	020032	104001			HLT	1	RX MODE REG NOT=7
2812	020034	112777	000006	161332	MOVB	#6,2DVSRSH	TX BCC REG
2813	020042	017704	161330		MOV	2DVSRSH,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	2DVSRSH	TXBCC
2818	020060	017704	161312		MOV	2DVSRSH,R4	
2819	020064	001402			BEQ	11\$	
2820	020066	005005			CLR	R5	
2821	020070	104001			HLT	1	RX BCC REG S/B=0
2822	020072	010005		11\$:	MOV	R0,R5	LOAD LINE NO.
2823	020074	000305			SWAB	R5	PUT IN HIGH BYTE
2824	020076	052705	050000		BIS	#BIT14+BIT12,R5	SET BCC COMPLETE
2825	020102	017704	161260		MOV	2DVSRSH,R4	READ RIC
2826	020106	020504			CMP	R5,R4	
2827	020110	001401			BEQ	12\$	
2828	020112	104001			HLT	1	DVRIC INCORRECT
2829	020114	104413		12\$:	RSMCLR	SCOP!	CLEAR ALL SEC REGS
2830	020116	104401			INC	R0	RETURN WITH SAME LINE
2831	020120	005200			DEC	R2	UPDATE LINE POINTER
2832	020122	005302			BEQ	13\$	4 LINES DONE?
2833	020124	001402			JMP	13\$	BR IF NO
2834	020126	000137	017504				JMP IF YES

2835 020132 000207
2836 020134 000 020
2837 020136 002 003
2838 020140 020 004
2839 020142 006 000

50\$: RTS PC ;EXIT
.BYTE 0,20
.BYTE 2,3
.BYTE 20,4
.BYTE 6,0

***** TEST 17 *****
*TEST OF RECEIVER RESYNC
*TEST TO TRANSMIT A BLOCK OF
*DATA (SYN, SYN, 1, 2, 3, 4, 5)
*HAVING CHAR "1" BEING A "SPECIAL CHAR" TO THE RECEIVER
*AT WHICH TIME A "RE-SYNC" PULSE WILL BE ISSUED
*AND A RESTART CHAR PROC. (DVSCROB=1) WILL BE DONE.
*WHEN THE TRANSMITTER IS DONE (DVSCRI5=1) A SECOND
*BLOCK OF DATA (SYN, SYN, SYN, SYN, 6, 7, 10)
*WILL BE SENT EXPECTING THAT THE NEXT TIME DVSCFJ7=1
*THAT THE DVRIC WILL HAVE:
*14=1 11:08=LINE NO. 07:00= "10"
*RXBUFFER (CORE) S/B= 1, 6, 7, 10.
*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.

: TEST 17

2859 020144 012737 000017 001226
2860 020152 012737 021142 001216
2861 020160 012700 000000
2862 020164 013737 001416 001236
2863 020172 100402
2864 020174 004737 020262
2865 020200 012700 000004
2866 020204 013737 001420 001236
2867 020212 100402
2868 020214 004737 020262
2869 020220 012700 000010
2870 020224 013737 001422 001236
2871 020232 100402
2872 020234 004737 020262
2873 020240 012700 000014
2874 020244 013737 001424 001236
2875 020252 100402
2876 020254 004737 020262
2877 020260 104400
2878 020262
2879 020262 012737 020314 001220
2880 020270 104413
2881 020272 112737 000001 030161
2882 020300 005037 030166
2883 020304 005037 030170
2884 020310 012702 000004
2885 020314 010077 161052
2886 020320 032737 004000 001236
2887 020326 001406
2888 020330 004537 022120
2889 020334 000 001
2890 020336 022560

ST17: MOV #17, TSTNO
MOV #TST20, NEXT
MOV #0, RO
MOV L00.03, STAT
BMT 100\$
JSR PC, 105\$
100\$: MOV #4, RO
MOV L04.07, STAT
BMT 101\$
JSR PC, 105\$
101\$: MOV #8, RO
MOV L08.11, STAT
BMT 102\$
JSR PC, 105\$
102\$: MOV #12, RO
MOV L12.15, STAT
BMT 103\$
JSR PC, 105\$
103\$: SCOPE
105\$: MOV #15, LOCK
RAMCLR
MOV #BIT0, RXTAB+1
CLR RXTAB+6
CLR RXTAB+10
MOV #4, R2
1\$: MOV RO, DVSR5
BIT #ASYNC, STAT
BEQ 80\$
PERFORM SETREG
.BYTE 000, 001
TXBAP

; PLACE LINE NUMBER INTO RO
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 1
; PLACE LINE NUMBER INTO RO
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; GO DO THE TEST FOR LINE CARD 2
; LOAD LINE NUMBER
; LOAD LINE CARD STATUS INTO STAT
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TEST FOR LINE CARD 3
; LOAD LINE NO.
; LOAD LINE CARD STATUS
; BR IF LINE CARD NOT TO BE TESTED
; DO THE TESTS FOR LINE CARD 4
; SCOPE THIS TEST.
; TEST ENTRANCE.
; SET RETURN
; CLEAR ALL SEC. REGS
; SET "SPECIAL CHAR" CNTRL BYTE
; CLEAR
; OTHER CNTRL BYTES
; SET FOR 4 LINE GROUP
; LOAD LINE NUMBER
; IS THIS AN ASYNC LINE CARD?
; BR IF NOT ASYNC.
; ADJUST FOR ASYNC LINE CARD
; REGISTERS
; LOAD FOR ASYNC

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

2947	020566	113725	001236		MOV B	STAT, (R5)+	: SYNC
2948	020572	012704	000006		MOV	#6, R4	: SET 1ST DATA TO 6
2949	020576	110425		3\$:	MOV B	R4, (R5)+	: LOAD
2950	020600	005204			INC	R4	: DATA
2951	020602	022704	000011		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		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
2965	020654	001403			BEQ	.+10	: ASYNC LINE CARD?
2966	020656	004537	022164		PERFORM	LOAD.MODE	: BR IF NO
2967	020662	000000			0		: CLEAR RX ENABLE
2968	020664	112777	000013	160502	MOV B	#13, ADVSRSH	: LINE STATE
2969	020672	042777	000200	160476	BIC	#BIT7, ADVSRA	: CLEAR "USE SEC TABLES"
2970	020700	052777	000002	160470	BIS	#BIT1, ADVSRA	: SET RE-SYNC
2971	020706	112777	000012	160460	MOV B	#12, ADVSRSH	: SEL LINE PROTOCOL PARAM.
2972	020714	052777	000002	160454	BIS	#BIT1, ADVSRA	: 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	032737	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 B	#13, ADVSRSH	: SEL LINE STATE.
2982	020762	052777	000004	160406	BIS	#BIT2, ADVSRA	: SET TX GO.
2983	020770	052777	000400	160364	BIS	#BIT8, ADVSCR	: RESTART CPU
2984	020776	005004			CLR	R4	: SET FOR TIME OUT.
2985	021000	105777	160356		TSTB	ADVSCR	: 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	ADVRC, 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
2999	021044	112304			MOV B	(R3)+, R4	: CHECK RX DATA
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			SCOPI			: 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

```

***** TEST 20 *****
*TEST OF RECEIVER OVERRUN.
*TEST TO TXMIT 134. CHARS AND RECV 129
*SERVICEING THE FIRST CHAR AS A SPECIAL CHAR
*AND STOPING THE CHAR PROCESSOR.
*WHEN THE TRANSMITTER FINISHES ALL 134. CHARS
*THE RECEIVER IS RESTARTED AND THE NEXT ENTRY
*IN THE RIC REG S/B OVER RUN ON CHAR 202(8).
*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
*****

```

3038	; TEST 20						
3039	-----						
3040	021142	012737	000020	001226	1ST20: MOV	#20,TSTNO	
3041	021150	012737	002436	001216	MOV	#.EOP,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	LO0.03,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		100\$: 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	LO4.07,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		101\$: MOV	#8.,R0	: LOAD LINE NUMBER
3053	021236	113737	001410	001244	MOV	MASK.C,MASKX	: GET MASK
3054	021244	013737	001422	001236	MOV	LO8.11,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		102\$: MOV	#12.,R0	: LOAD LINE NO.
3058	021264	113737	001411	001244	MOV	MASK.D,MASKX	: GET MASKK

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						:TEST ENTRANCE.
3064	021310	012737	021346	001220	MOV	#1\$,LOCK	:RETURN FOR SW09
3065	021316	104413			RAMCLR		:CLEAR ALL SEC REGISTERS
3066	021320	005004			CLR	R4	:CLEAR
3067	021322	012705	030160		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	030161	MOVB	#BIT0,RXTAB+1	:SET "SPECIAL CHAR"(1)
3072	021342	012702	000004		MOV	#4,R2	:4 LINE GROUP
3073	021346	010077	160020		MOV	RO,DOVSRS	: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	000	001		.BYTE	000,001	:REGISTERS
3078	021370	022560			TXBAP		:LOAD FOR ASYNC
3079	021372	177573			-133.		:LOAD FOR ASYNC
3080	021374	000405			BR	81\$	:CONTINUE TEST
3081	021376	004537	022120		PERFORM	SETREG	
3082	021402	000	001		.BYTE	000,001	:TX BA P, TX BC P
3083	021404	022556			SYNC		
3084	021406	177572			-134.		
3085	021410						
3086	021410	032737	004000	001236	BIT	#ASYNC,STAT	:IS THIS AN ASYNC LINE CARD?
3087	021416	001406			BEQ	82\$	:BR IF NOT ASYNC.
3088	021420	004537	022120		PERFORM	SETREG	:ADJUST FOR ASYNC LINE CARD
3089	021424	004	005		.BYTE	004,005	:REGISTERS
3090	021426	027560			RXBA		:LOAD FOR ASYNC
3091	021430	177576			-130.		:LOAD FOR ASYNC
3092	021432	000405			BR	83\$	:CONTINUE TEST
3093	021434	004537	022120		PERFORM	SETREG	
3094	021440	004	005		.BYTE	004,005	:RX BA, RX BC
3095	021442	027560			RXBA		
3096	021444	177577			-129.		
3097	021446	004537	022120		PERFORM	SETREG	
3098	021452	010	011		.BYTE	010,011	:TX TAB, RX TAB
3099	021454	023560			TXTAB		
3100	021456	030160			RXTAB		
3101	021460	004537	022120		PERFORM	SETREG	
3102	021464	013	012		.BYTE	013,012	:LINE STATE, LINE PROTOCCL PARAM
3103	021466	000004			BIT2		:TX GO
3104	021470	000101			BIT6+BIT0		:TX ODCMP + IDLE MARK
3105	021472	032737	004000	001236	BIT	#ASYNC,STAT	:IS THIS ASYNC LINE CARD?
3106	021500	001412			BEQ	60\$	:BR IF NO.
3107	021502	004537	022164		PERFORM	LOAD.MODE	:LOAD PARAMETERS.
3108	021506	020000			BIT13		:RECEIVER ENABLE
3109	021510	004537	022164		PERFORM	LOAD.MODE	
3110	021514	015000			<BIT12+BIT11>+BIT9		:#8 BITS/PER/CHAR
3111	021516	004537	022164		PERFORM	LOAD.MODE	
3112	021522	072000			<BIT14+BIT13+BIT12>+BIT10		:#9600 BAUD.
3113							
3114	021524	000403			BR	61\$	

3115	021526	004537	022164	60\$:	PERFORM	LOAD,MODE	:LOAD
3116	021532	034000			BIT13+BIT12+BIT11		:MODE
3117	021534	012705	022560	61\$:	MOV	#TXBAP,F5	:LOAD
3118	021540	005004			CLR	R4	:TX
3119	021542	105204		2\$:	INCB	R4	:DATA
3120	021544	001402			BEQ	21\$	:BUFFER
3121	021546	110425			MOVB	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	MOVB	#12,DVSRSH	:LINE PROTOCOL PARAM.
3130	021604	052777	000040	157564	BIS	#BIT5,DVSRSH	: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	DVSRIC,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	104001			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	+6	:BR IF YES
3151	021700	000137	021346		JMP	1\$	:JMP IF YES
3152	021704	000207			RTS	PC	:EXIT
3153							
3154	021706			SETSYNC:			
3155	021706	113737	001236	022556	MOVB	STAT,SYNC	:SET SYNC FOR THIS LINE.
3156	021714	113737	022556	022557	MOVB	SYNC,SYNC+1	:PLACE SYNC IN HIGH BYTE
3157	021722	032737	010000	001236	BIT	#TWO\$YN,STAT	:ONE SYNC OR TWO?
3158	021730	001402			BEQ	1\$	:BR IF JUMPERED 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		BIC	R0,BCCFBK
3177	022024	000241			CLC	
3178	022026	006037	001252		ROR	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			BIS	R2,R0
3185	022054	043737	022112	001252	BIC	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			LRC8=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,ADVSRSH
3207	022134	012577	157236		MOV	(R5)+,ADVSRSH
3208	022140	042777	000060	157214	BIC	#BITS+BIT4,ADVSCR
3209	022146	110177	157222		MOVB	R1,ADVSRSH
3210	022152	012577	157220		MOV	(R5)+,ADVSRSH
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)+,ADVLCR
3217	022170	052777	100000	157172	BIS	#BIT15,ADVLCR
3218	022176	010046			MOV	R0,-(SP)
3219	022200	005000			CLR	R0
3220	022202	005777	157162	1\$:	TST	ADVLCR
3221	022206	100004			BPL	2\$
3222	022210	104414			DELAY	
3223	022212	005200			INC	R0
3224	022214	001372			BNE	1\$
3225	022216	104000			HLT	0
3226	022220	012600		2\$:	MOV	(SP)+,R0

;BIT 15 FAILED TO CLEAR

```

3227 022222 000205 EXIT
3228
3229 :SUBROUTINE.
3230 :CORE TABLES ALREADY SET UP
3231 :XMIT 3 CHARS 2SYNC+ 1 DATA
3232 :RCV 1 CHAR
3233 022224 010077 157142 DV110N: MOV R0, DVSR5
3234 022230 032737 004000 001236 BIT #ASYNC, STAT ;#IS THIS AN ASYNC LINE CARD?
3235 022236 001406 BEQ 80$ ;#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 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$: PERFORM ,LOAD.MODE
3268 022364 034000 BIT13+BIT12+BIT11
3269 022366 004537 021706 PERFORM SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3270 022372 000207 61$: RTS PC
3271
3272
3273 SETSCAN:
3274 022374 010346 MOV R3, -(SP)
3275 022376 052777 000010 156756 BIS #BIT3, DVSCR
3276 022404 012503 MOV (R5)+, R3
3277 022406 001414 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:  .BLKW 1
3294 022560 000400      TXBAP:  .BLKB 400
3295 023160 000400      TXBAS:  .BLKB 400
3296 023560 000400      TXTAB:  .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 032560 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
      .BYTE 6,4
      SAVR5
      .BYTE 6,2
      SAVR4
      .BYTE 2,1
      SAVR0
3315 034366 000003
3316 034370 006 004
3317 034372 001272
3318 034374 006 002
3319 034376 001270
3320 034400 002 001
3321 034402 001260
3322 034404
3323 034404 000000      .ERRTAB: 0
3324 034406 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

```

K06

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 65
CZDVDC.P11 02-FEB-78 13:55

DV11 DEVICE DIAGNOSTICS.

COPYRIGHT 1975 DIGITAL EQUIP. CORP.

SEQ 0075

3333 034430 034333
3334 034432 034366
3335 034434 034307
3336 034436 034333
3337 034440 034366
3338
3339 034442
3340 000001

DH1 ;HALT 3
DT1
EM4
DH1 ;HALT 4
DT1

::*****
CORMAX:
.END

```
ADRCNT= 003443
ALU    = 010000
ASync  = 004000
```

```
AUTO.S 006622
BCC = 060000
BCCFBK 022114
BINWRD 003746
BITO = 000001
```

BIT1 = 000002

BIT10 = 002000

BIT11 = 004000

BIT12 = 010000

BIT13 = 020000

BIT14 = 040000

BIT15 = 100000

EIT2 = 000004

BIT3 = 000010

BIT4 = 000020

BITS = 000040

BIT6 = 000100

BIT7 = 000200

BIT8 = 000400

SIT9 = 001000

BRB = 070000

BRW - 003014

BRX 003016

CALBCC 022116

CHRCNT 003744

CLKX 001242

CLK.A	001412
CLK.B	001412

CLK.B	001413
CLK.C	001414

CLK.C	001414
CLK.D	001415

CLK.D 001415

617*	653*	662*										
73#												
81#	1260	1338	1357	1380	1462	1481	1504	1615	1627	1646	1678	2046
2277	2304	2417	2444	2596	2619	2744	2767	2886	2909	2953	2964	2977
3074	3086	3105	3138	3234	3257							
1127#												
78#												
3167*	3173*	3176*	3179	3183	3195#							
703*	704	741#										
71#	884	1273	1503	1645	2068	2143	2156	2298	2329	2438	2469	2766
2801	2808	2881	2908	3071	3104	3171	3256					
70#	884	895	1914	2068	2329	2469	2549	2723	2725	2801	2808	2970
2972	3278	3281										
61#	884	1265	1387	1511	1653	2299	2302	2311	2439	2442	2451	2564*
2565*	2566*	2567*	2568*	2574*	2575*	2576*	2577*	2578*	2626	2774	2916	3112
3264												
60#	884	1263	1269	1385	1391	1509	1515	1651	1657	2309	2315	2449
2455	2624	2630	2772	2778	2914	2920	3110	3116	3262	3268		
59#	73	75	77	79	1263	1265	1269	1385	1387	1391	1509	1511
1515	1651	1653	1657	2309	2311	2315	2323	2449	2451	2455	2463	2624
2626	2630	2772	2774	2778	2824	2914	2916	2920	3110	3112	3116	3262
3264	3268	3278	3281									
58#	74	75	78	79	1265	1273	1281	1383	1387	1391	1507	1511
1515	1649	1653	1657	2299	2302	2307	2311	2315	2439	2442	2447	2451
2455	2622	2626	2630	2770	2774	2778	2912	2916	2920	2980	3108	3112
3116	3137	3260	3264	3268								
57#	76	77	78	79	524	1265	1387	1511	1653	2186	2299	2302
2311	2323	2439	2442	2451	2463	2626	2774	2824	2916	2994	3112	3264
3278	3281											
56#	2282	2287	2291	2299	2302	2422	2427	2431	2439	2442	3217	
69#	451	884	1254	1378	1502	1644	2299	2329	2439	2469	2618	2728
2737	2765	2801	2808	2907	2982	3103	3255					
68#	884	1738	1770	1792	2257	2273	2298	2397	2413	2550	2719	2737
2766	3275											
67#	1607	1608	1792	2298	2731	2736	2766	3208				
66#	1293	1869	1975	1979	2068	2560	2562	2564	2566	2567	2568	2573
2572	2574	2576	2577	2578	2728	2737	3130	3208				
65#	1195	1975	1979	2068	2152	2156	2179	2561	2562	2565	2566	2567
2568	2571	2572	2575	2576	2577	2578	2728	2737	2908	3104	3278	3281
64#	518	765	922	943	1195	1975	2563	2564	2565	2566	2567	2568
2573	2574	2575	2576	2577	2578	2728	2737					

M06

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 68

SEG 0077

CZDVDC.P11 02-FEB-78 13:55

CROSS REFERENCE TABLE -- USER SYMBOLS

CNVRT =	104411	209*	476	478		480	482	802	804	860	917				
CONVRT=	104410	207*	416	818											
CORMAX	034442	3339#	3340												
CRC.CC=	102010	1799	3199#												
CRC16 =	120001	1777	3198#												
CREAM	001306	171#	385*	992*	993	995*	1000	1001*	1002	1005*					
CSRMAP	006624	410	1129#												
CYCLE	005666	460	496	497	982#										
DATA	034160	3313#													
DATABP	004276	791*	794	816	819#										
DATACL=	104416	219#													
DataMD	004264	790*	812	815#											
DELAY =	104414	215#	1275	1663	2161	2934	2941	2987	3222						
DEVAOR	003440	615*	650	660#											
DH1	034333	3314#	3327	3330	3333	3336									
DT1	034366	3314#	3328	3331	3334	3337									
DVACTV	001300	165#	432*	433	982	987	1161*	1167*	1168*	1172	1191				
DVCROO	001500	281#													
DVCRO1	001524	292#													
DVCRO2	001550	303#													
DVCRO3	001574	314#													
DVCRO4	001620	325#													
DVCRO5	001644	336#													
DVCRO6	001670	347#													
DVCRO7	001714	358#													
DVLCR	001370	233#	901*	902	1021*	1022*	1023	3216*	3217*	3220					
DVNSR	001402	238#	1031*	1032*	1033										
DVNUM	001301	166#	381	486	1133*	1154*	1155	1162	1164						
DVRIC	001366	232#	1019*	1020*	1021	2169	2187	2320	2460	2825	2991	3134			
DVRLVL	001354	227#	1036*	1037*	1038	2155*									
DVRVEC	001352	226#	503	1007*	1036	2154*									
DVSCR	001362	230#	500	878*	891*	895*	1006*	1017	1273*	1281*	1393*	1394	1517*	1518	
		1659*	1661	1749*	1750	1771*	1772	1793*	1794	1858*	1859	1924*	1925	1981*	
		1982	2069*	2070	2156*	2173*	2179*	2182	2317*	2318	2457*	2458	2631*	2632	
		2786*	2787	2930*	2932	2939	2983*	2985	3124*	3125	3127	3131*	3132	3208*	
		3275*													
DVSCRH	001364	231#	1017*	1018*	1019										
DVSFR	001400	237#	1029*	1030*	1031	3278*	3281*								
DVSRA	001376	236#	882	1027*	1028*	1029	1285	1753	1775	1797	1862	1934*	1979*	1986	
		1990	1994	1998	2068*	2152*	2328	2334	2340	2344	2468	2474	2		

N06

SEQ 0078

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 69
 CZDVDC.P11 02-FEB-78 13:55 CROSS REFERENCE TABLE -- USER SYMBOLS

DVTVEC	001356	228#	1038*	1039*	1040	1271*				
DV.END	001740	369#	993	1002	1131					
DV.MAP	001500	171	280#	385	412	995	1005	1129	1134	1183
DV00.A	001504	283#								
DV00.B	001510	285#								
DV00.C	001514	287#								
DV00.D	001520	289#								
DV01.A	001530	294#								
DV01.B	001534	296#								
DV01.C	001540	298#								
DV01.D	001544	300#								
DV02.A	001554	305#								
DV02.B	001560	307#								
DV02.C	001564	309#								
DV02.D	001570	311#								
DV03.A	001600	316#								
DV03.B	001604	318#								
DV03.C	001610	320#								
DV03.D	001614	322#								
DV04.A	001624	327#								
DV04.B	001630	329#								
DV04.C	001634	331#								
DV04.D	001640	333#								
DV05.A	001650	338#								
DV05.B	001654	340#								
DV05.C	001660	342#								
DV05.D	001664	344#								
DV06.A	001674	349#								
DV06.B	001700	351#								
DV06.C	001704	353#								
DV06.D	001710	355#								
DV07.A	001720	360#								
DV07.B	001724	362#								
DV07.C	001730	364#								
DV07.D	001734	366#								
DV110N	022224	1744	1766	1788	1853	1918	1977	2060	2146	3233#
EM1	034162	3314#	3326							
EM2	034212	3314#	3329							
EM3	034247	3314#	3332							
EM4	034307	3314#	3335							
ERRCNT	001232	142#	387*	509	827*					
ERRFLG	001311	177#	383*	471*	538*	779*	792	806*	861*	
ERRMSG	004252	789#	807	810#						
ERTAB0	004366	804	836#							
EXIT =	000205	81#	3160	3213	3227	3290				
EXITER	004322	822	827#							
FIX.00	006516	1046	1051	1056	1061	1095#				
HALTS	004302	775	821#							
HILIM	003436	614*	641	659#						
ICOUNT	001222	138#	536	541*						
INBUF	005520	584	620	967#						
INIFLG	001310	176#	392	407*						
INSTER=	104404	199#	635							
INSTR =	104403	197#	1068							
INSTR2	003236	591	603#							
LIGHT	000174	110#	402							

LIGHTS	001200	121#	402*	473*										
LIMITS	003364	630	641#											
LOAD.M	022164	1262	1264	1268	1382	1384	1386	1390	1506	1508	1510	1514	1648	1650
		1652	1656	2306	2308	2310	2314	2446	2448	2450	2454	2621	2623	2625
		2629	2769	2771	2773	2777	2911	2913	2915	2919	2966	2979	3107	3109
		3111	3115	3215#	3259	3261	3263	3267						
LOBITS	003442	616*	645	661#	662									
LOCK	001220	137#	540*	554	556	798	1240*	1341*	1465*	1595*	1735*	1845*	1912*	1972*
		2049*	2131*	2249*	2392*	2544*	2707*	2879*	3064*					
LOGICA	002560	107	490#											
LOKFLG	001312	178#												
LOLIM	003434	613*	643	658#										
LPCNT	001224	139#	535*	536	539*									
LRCB =	000200	1755	3197#											
LSTERR	001234	143#	388*	470*	522*	776	778*	862*						
LOO.03	001416	255#	1008*	1043	1223	1318	1442	1575	1718	1828	1895	1955	2026	2114
		2232	2375	2527	2690	2862	3044							
LO4.07	001420	256#	1010*	1048	1227	1323	1447	1580	1722	1832	1899	1959	2031	2118
		2236	2379	2531	2694	2866	3049							
LO8.11	001422	257#	1012*	1053	1231	1328	1452	1585	1726	1836	1903	1963	2036	2122
		2240	2383	2535	2698	2870	3054							
L12.15	001424	258#	1014*	1058	1235	1333	1457	1590	1730	1840	1907	1967	2041	2126
		2244	2387	2539	2702	2874	3059							
MASKX	001244	151#	1317*	1322*	1327*	1332*	1410	1441*	1446*	1451*	1456*	1534	1574*	1579*
		1584*	1589*	1683	2025*	2030*	2035*	2040*	2080	3043*	3048*	3053*	3058*	3142
MASK.A	001406	245#	1044	1317	1441	1574	2025	3043						
MASK.B	001407	246#	1049	1322	1446	1579	2030	3048						
MASK.C	001410	247#	1054	1327	1451	1584	2035	3053						
MASK.D	001411	248#	1059	1332	1456	1589	2040	3058						
MASTEK	005400	800	958#											
MCRLF	005104	569	692	796	797	805	946	958#	1067	1085				
MCSRX	005330	475	958#											
MDATA	005624	719	729	971#										
MEPASS	005145	474	958#											
MERRPC	005454	803	958#											
MERRX	005355	481	958#											
MERR2	005174	958#	984	1174										
MERR3	005243	429	958#											
MLOCK	005301	453	958#											
MNEW	005402	424	958#											
MPASSX	005344	479	958#											
MPFAIL	005107	859	958#											
MQM	005100	599	958#	1090										
MR	005171	461	958#											
MRESET=	004000	81#	878	891										
MSTCLR=	104412	211#	863	1290	1420	1544	1692	1742	1764	1786	1867	1932	2001	2091
		2201	2662	3019	3146									
MTITLE	001000	119#	406											
MTSTN	005366	801	958#	1069										
MTSTPC	005267	958#												
MVECX	005336	477	958#											
NEXT	001216	136#	542	832	1221*	1315*	1439*	1572*	1716*	1826*	1893*	1953*	2023*	2112*
		2230*	2373*	2525*	2688*	2860*	3041*							
NOLIST=	***** U	1												
NPR =	040000	76#												
PARAM =	104405	201#	1070											

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 71
CZDVDC.P11 02-FEB-78 13:55 CROSS REFERENCE TABLE -- USER SYMBOLS

```
PARAM1      003304
PARBIT=     040000
PARERR      003360
PASCNT      001230
PERFOR=     004537
```

[illegible]

```
PFTAB      004470
POPRO =    012600
POP1SP=    005726
POP2SP=    022626
PS         =    177776
PUSHRO=    010046
PUSH1S=    005746
PUSH2S=    024646
QV.FLG     001313
RAM        =    020000
RAMCLR=    104413
```

860	866											
55	826											
55												
55	544											
53	378*	447*	1194*	1270*	1276*	1282*	2153*	2160*				
55	823											
55												
55												
179	384*	485*	533									
74												
213	864	1241	1342	1466	1596	1736	1846	1913	1973	2050	2132	2253

```

REGBUF      022452
RESREG      004300
RESTAR      004414
RESTR       002572
RESV16      001404
RES05 =     104407
RETURN      001214
ROMCLK=     104415
RUN          001304
RXBA        027560

```

3292#	817	847	484	239#	205#	135#	217#	169#	1354*
820#	853#	488	1033*	820	390*	3279	386*	1355*	
		496#	1034*		460*	3282	987	1356*	
					462	990*	990*	1370	
					496*	991*	991*	1398	
					542*	998*	998*	1403	
					545	999*	999*	1412	
					832*	1478*	1478*	1478*	
					834	1479*	1479*	1479*	
					865	1480*	1480*	1480*	
					1084*	1494	1494	1494	
					1092*	1522	1522	1522	
					1093	1527	1527	1527	

RXTAB 030160

1344	1375	1468	1499	1607*	1608*	1609*	1610*	1641	1738*	1852*	1916*	1976*
------	------	------	------	-------	-------	-------	-------	------	-------	-------	-------	-------

```

SAVACT      001302
SAVNUM      001303
SAVPC       001276
SAVR0       001260
SAVR1       001262
SAVR2       001264
SAVR3       001266
SAVR4       001270
SAVR5       001272
SAVSP       001274
SAV05      = 104406
SCOPE      = 104400

```

167#	381*	483*	486*	1165*								
164#	667*	838										
157#	676*	681	3320									
158#	675*	682										
159#	674*	683										
160#	673*	684										
161#	672*	685	3318									
162#	671*	686	3316									
163#												
203#	780											
191#	1238	1336	1460	1593	1733	1843	1910	1970	2044	2129	2247	2390

SEQ 0081

CROSS REFERENCE TABLE -- USER SYMBOLS

1938	1291	1421	1545	1693	1808	1874	1935	2002	2092	2202	2348	2488
------	------	------	------	------	------	------	------	------	------	------	------	------

[illegible]

3249	3253											
3273#												
1392	1516	1658	2316	2456	2553	2779	2929	3123	3154#	3269		
1756	1778	1800	3161#									
702*	723	726*	740#									
112#	403											
54#	379	448	833	855								
148#	1223*	1227*	1231*	1235*	1260	1318*	1323*	1328*	1333*	1338	1357	1380
1408	1442*	1447*	1452*	1457*	1462	1481	1504	1575*	1580*	1585*	1590*	1615
1627	1646	1678	1718*	1722*	1726*	1730*	1828*	1832*	1836*	1840*	1895*	1899
1903*	1907*	1955*	1959*	1963*	1967*	2026*	2031*	2036*	2041*	2046	2054	2058
2059	2078	2114*	2118*	2122*	2126*	2232*	2236*	2240*	2244*	2250	2261	2277
2304	2375*	2379*	2383*	2387*	2401	2417	2444	2527*	2531*	2535*	2539*	2555
2596	2619	2690*	2694*	2698*	2702*	2708	2744	2767	2862*	2866*	2870*	2874
2886	2909	2946	2947	2953	2964	2977	3044*	3049*	3054*	3059*	3074	3086
3105	3133	3155	3157	3234	3257							

671#												
123#	398	403*	408	422	427	432	451	515	524	531	552	565
762	769	774	821	828	830	914	915*	925*	930*	931*	932*	939

1062			
34	422		
33	1062		
32			
31			
30			
29			
28			
27			
26	828		
25	552		
24	830		
23	531		
22	565	769	
21	774		
20			
19			
284			
295			
306			
317			
328			
339			
350			
361			
286			
297			
308			
319			

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 73
CZDVDC.P11 02-FEB-78 13:55 CROSS REFERENCE TABLE -- USER SYMBOLS

SYN804	001632	330#												
SYN805	001656	341#												
SYN806	001702	352#												
SYN807	001726	363#												
SYNC	022556	1366 3243	1490 3293#	1624	2286	2426	2605	2753	2895	2962	3083	3155*	3156*	3159*
SYNCX	001240	149#												
SYNC00	001516	288#												
SYNC01	001542	299#												
SYNC02	001566	310#												
SYNC03	001612	321#												
SYNC04	001636	332#												
SYNC05	001662	343#												
SYNC06	001706	354#												
SYNC07	001732	365#												
SYNC2A	001426	260#	1009*											
SYNC2B	001430	261#	1011*											
SYNC2C	001432	262#	1013*											
SYNC2D	001434	263#	1015*											
SYND00	001522	290#												
SYND01	001546	301#												
SYND02	001572	312#												
SYND03	001616	323#												
SYND04	001642	334#												
SYND05	001666	345#												
SYND06	001712	356#												
SYND07	001736	367#												
S.C	= 050000	77#												
TEMP	005562	706	856*	857*	969#									
TEMP1	001246	152#	412*	413	418*	960	1191*	1192*	2636*	2644*	3164*	3187*		
TEMP2	001250	153#	413*	414	962	3165*	3169*							
TEMP3	001252	154#	2791*	2797*	3166*	3168	3178*	3180	3185*	3186*	3189			
TEMP4	001254	155#												
TEMP5	001256	156#												
TKCSR	001204	127#	527	586	912	919	940							
TKDBR	001206	128#	517	529	588	594	764	921	942					
TLAST	= 021142	1088	3314#											
TPCSR	001210	129#	570	592	771	926								
TPDBR	001212	130#	572*	594*	773*	928*								
TRPOK	003762	752#												
TSTNO	001226	140#	389*	841	868	1073	1080	1082	1220*	1314*	1438*	1571*	1715*	1825*
TST1	007256	1892*	1952*	2022*	2111*	2229*	2372*	2524*	2687*	2859*	3040*			
TST10	013244	1076	1092	1220#										
TST11	013554	1893	1952#											
TST12	014166	1953	2022#											
TST13	014650	2023	2111#											
TST14	015442	2112	2229#		</									

F07

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 74
CZDVDC.P11 02-FEB-78 13:55 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0083

TST5	012004	1572	1715*											
TST6	012512	1716	1825*											
TST7	012772	1826	1892*											
TTST	002702	454*	455*	457*	458*	525*								
TWOSYN=	010000	81*	3157											
TXBAP	022560	1244*	1250	1350*	1351*	1361	1474*	1475*	1485	1602	1619	1739*	1850*	1915*
		1980*	2053	2136	2265	2261	2405	2421	2585	2600	2634	2748	2781	2890
		2923	2945	2957	3078	3117	3238	3294*						
TXBAS	023160	3295*												
TXTAB	023560	1246*	1255	1343	1374	1467	1498	1597*	1598*	1599*	1600*	1601*	1640	1737*
		1851*	1914*	1975*	1976	2256	2260	2294	2396	2400	2434	2546	2557	2560*
		2561*	2562*	2563*	2564*	2565*	2566*	2567*	2568*	2613	2712	2720*	2721*	2722*
		2723*	2724*	2725*	2726*	2727*	2728*	2761	2903	3099	3251	3296*		
TYPDAT	004266	795	813	816*										
TYPE =	104402	195*	406	411	424	429	453	461	474	475	477	479	481	569
		582	599	692	729	796	797	800	801	803	805	809	814	859
		916	918	946	984	1067	1085	1090	1174					
TYPMSG	004166	793	796*											
VECMAP	007102	1173	1181*											
WRDCNT	003742	700*	730*	738*										
WRKO.F	004254	808	811*											
XBX	004060	770	772	774*										
XCSR	002604	476	498*											
XERR	002626	482	507*											
XFR =	030000	75*												
XHEAD	005461	411	958*											
XPASS	002620	480	504*											
XPOLY	022112	1755*	1777*	1799*	3174	3185	3194*							
XSTATQ	005506	417	958*											
XTSTN	004374	802	839*											
XVEC	002612	478	501*											
SCRAP =	177777	1*	1212*	1216*	1304*	1310*	1428*	1434*	1552*	1567*	1700*	1711*	1815*	1821*
		1881*	1888*	1942*	1948*	2009*	2018*	2099*	2107*	2210*	2225*	2355*	2368*	2495*
SE =	000022	2520*	2669*	2683*	2841*	2855*	3027*	3036*						
		1*	1221	1222*	1315	1316*	1439	1440*	1572	1573*	1716	1717*	1826	1827*
		1893	1894*	1953	1954*	2023	2024*	2112	2113*	2230	2231*	2373	2374*	2525
SN =	000020	2526*	2688	2689*	2860	2861*	3041	3042*						
		1*	1212	1218	1222*	1304	1312	1316*	1428	1436	1440*	1552	1569	1573*
		1700	1713	1717*	1815	1823	1827*	1881	1890	1894*	1942	1950	1954*	2009
		2020	2024*	2099	2109	2113*	2210	2227	2231*	2355	2370	2374*	2495	2522
		2526*	2669	2685	2689*	2841	2857	2861*	3027	3038	3042*	3314*		
Si =	000017	1*	182*	191	193*	195*	197*	199*	201*	203*	205*	207*	209*	211*
		213*	215*	217*	219*	221*								
.	= 034442	92*	93	96*	103*	104*	105*	106*	109*	111*	114*	118*	120*	165*
		166*	167*	168*	169*	170*	279*	281*	282*	283*	284*	285*	286*	287*
		288*	289*	290*	292*	293*	294*	295*	296*	297*	298*	299*	300*	301*
		303*	304*	305*	306*	307*	308*	309*	310*	311*	312*	314*	315*	316*
		317*	318*	319*	320*	321*	322*	323*	325*	326*	327*	328*	329*	330*
		331*	332*	333*	334*	336*	337*	338*	339*	340*	341*	342*	343*	344*
		345*	347*	348*	349*	350*	351*	352*	353*	354*	355*	356*	358*	359*
		360*	361*	362*	363*	364*	365*	366*	367*	431	520	767	849	858
		872	910*	920	927	941	968*	970*	972*	986	1177	1198	1339	1395
		1463	1519	1762	1784	1806	1860	1926	1983	1987	2047	2071	2163	2183
		2251	2319	2459	2633	2709	2788	2833	2933	2936	2940	2943	2965	2976
		2978	2986	2989	3023	3070	3126	3128	3133	3139	3150	3292*	3293*	3294*
		3295*	3296*	3297*	3298*	3299*	3300*	3301*	3302*	3303*	3304*	3305*	3306*	3307*

G07

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 75
 CZDVDC.P11 02-FEB-78 13:55 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0024

	3308#	3309#	3310#	3311#	3312#
.BEGIN 002332	447#				
.CNVRT 003542	210	693#			
.CONVR 003536	208	692#			
.DATAC 004576	220	898#			
.DELAY 004476	216	869#			
.EOP 002436	469#	3041			
.ERRTA 034404	788	3322#			
.HLT 004002	99	761#			
.INSTE 003224	200	599#			
.INSTR 003120	198	578#			
.INST1 003140	582#	602			
.MSC 003142	580#	583#			
.MSTCL 004556	212	890#			
.PARAM 003244	202	610#			
.PFAIL 004402	97	380	846#	854	
.RAMCL 004516	214	877#			
.RESOS 003504	206	681#			
.ROMCL 004566	218	894#			
.SAVOS 003444	204	667#			
.SCOPE 002634	192	514#			
.SCOP1 003020	194	552#			
.START 001742	115	378#	390		
.TRPSR 003750	101	749#			
.TRPTA 001314	190#	754			
.TYPE 003044	196	562#			

H07

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 77

SEQ 0085

CZDVDC.P1: 02-FEB-78 13:55 CROSS REFERENCE TABLE -- MACRO NAMES

OVEND	1#	463													
OVFRNT	1#														
HLT	55#	907	1279	1289	1401	1406	1417	1525	1530	1541	1666	1672	1677	1687	1631
	1763	1785	1807	1866	1931	1988	1992	1996	2000	2077	2085	2090	2164	2172	2190
	2197	2326	2332	2337	2342	2346	2466	2472	2477	2482	2486	2643	2650	2656	2661
	2796	2805	2811	2816	2821	2828	2937	2944	2990	2997	3003	3008	3013	3018	3145
	3225														
\$ADJUS	1#	1357	1481	1615	1627	2277	2417	2596	2744	2886	2953	3074	3086	3234	
\$BEAK	1#	1742	1764	1786											
\$BUFFE	1#	964													
\$CK15	1#														
\$CK150	1#														
\$CLR.T	1#														
\$CYCLE	1#	973													
\$EOP	1#	463													
\$FINI	1#	3314													
\$GETFL	1#														
\$GETPA	1#	1068													
\$HEADE	1#														
\$LC16	1#	1211													
\$LC16A	1#	1303	1427												
\$LC17	1#	1551													
\$LC18	1#	1699													
\$LC19	1#	1941													
\$LC20	1#	2008													
\$LC21	1#	2098													
\$LC22	1#	2209	2354												
\$LC30	1#	2494													
\$LC31	1#	2669													
\$LC32	1#	2840													
\$LC33	1#	3026													
\$MSG	1#	958													
\$PFAIL	1#	842													
\$RAMCL	1#	869													
\$RXSHI	1#														
\$SCOPE	1#	510													
\$SETAS	1#	1260	1380	1504	1646	2304	2444	2619	2767	2909	3106	3257	2370	2522	2685
\$SETLI	1#	1218	1312	1436	1569	1713	1823	1890	1950	2020	2107	2227			
	2857	3038													
\$SETSC	1#	3272													
\$SETSY	1#	3154													
\$SET.T	1#														
\$SIL01	1#														
\$SIMBC	1#	3161													
\$TRPDE	1#	191	193	195	197	199	201	203	205	207	209	211	213	215	217
	219														
\$TSTN	1#	1218	1312	1436	1569	1713	1823	1890	1950	2020	2109	2227	2370	2522	2685
	2857	3038													
\$TXSHI	1#														
\$VARIA	1#	117													
\$XZ	1#	1212	1216	1304	1310	1428	1434	1552	1567	1700	1711	1815	1821	1881	1888
	1942	1948	2009	2018	2099	2107	2210	2225	2355	2368	2495	2520	2669	2683	2841
	2855	3027	3036												

. ABS. 034442 000

CZDVDCO MACY11 30A(1052) 10-FEB-78 15:44 PAGE 78
CZDVDC.P11 02-FEB-78 13:55 CROSS REFERENCE TABLE -- MACRO NAMES

SEG 0086

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)