DV11

BASIC RIW + ROM INSTRUCTION

MD-11-DZDVA-B

EP DZDVA B DL A

OCT 1976

FICHE 1 OF 1

digital

Made In U.S.A.

.REM $

[illegible]

DESCRIPTION

MODULE CODE: MAINDEC-11-D2DA-B-D

MODULE NAME: BASIC B A TEST ARC BUS INSTRUCTION DIAGNOSTIC

DATE RELEASED: 21-APRIL-75

MAINTAINER: DIAGNOSTICS

AUTHOR: JOHN EGOLF

C01

1. ABSTRACT

THE FUNCTION OF THE [illegible] DIAGNOSTICS ARE TO VERIFY THAT THE [illegible] OPERATES ACCORDING TO SPECIFICATIONS. THE [illegible] ARE NO [illegible] FUNCTIONS AND THE ALL OPERATIONS OF THE [illegible] ARE CORRECT IN [illegible] ENVIRONMENT.
PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE [illegible] CONFIGURATION [illegible] THE "[illegible]" PROGRAM [illegible] [illegible] AND THEN EACH DIAGNOSTIC WILL [illegible] OVERLAY [illegible] ARE STORED IN THE "STATUS TABLE" SEE SECTION [illegible]. THE ALTERNATIVE [illegible] PROGRAM IS "AUTO SIZING" SEE SECTION [illegible].

[illegible] DOES R/W TESTS ON BOTH PRIMARY REGISTERS AND [illegible] SECONDARY REGISTERS. TESTS ARE MADE TO VERIFY THAT NO INTERACTION BETWEEN SECONDARY REGISTERS OF ANY LINES EXISTS. [illegible] ALSO EXERCISES [illegible] [illegible] INSTRUCTIONS AND VERIFIES INTERNAL REGISTERS [illegible] PROCESSOR. INTERUPTS AND NPRS ARE ALSO TESTED IN [illegible]
NOTE: FOR EASE OF DIAGNOSIS, ALL [illegible] LINE CARDS MAY BE PULLED [illegible] THE SYSTEM UNIT AS MAY THE TWO MODEM CONTROL [illegible]. ALSO THE DIAGNOSTIC IN NO WAY READS OR USES THE ROMS THAT ARE IN THE [illegible].

CURRENTLY THERE ARE SIX OFF LINE DIAGNOSTICS THAT ARE TO BE RUN IN [illegible] TO INSURE THAT IF AN ERROR [illegible] AN EARLY STAGE AND INSURING THAT DIAGNOSIS OF ERROR WILL BE IMMEDIATE [illegible] PROBLEM.
NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE SIX DIAGNOSTICS ARE:
1. [illegible] (REV [illegible]) BASIS R/W TEST AND ROM INSTRUCTION EXERCISER.
2. [illegible] (REV [illegible]) STATIC LINE CARD TESTS.
3. [illegible] (REV [illegible]) "FREE RUNNING" ROM TESTS PART 1.
4. [illegible] (REV [illegible]) "FREE RUNNING" ROM TESTS PART 2.
5. [illegible] (REV [illegible]) MODEM CONTROL AND [illegible] TESTS PLUS [illegible] PARAMETER [illegible] PROGRAM)
6. [illegible] (REV [illegible]) ASYNCHRONOUS LINE CARD TESTS.

2. REQUIREMENTS

2.1 EQUIPMENT

[illegible] FAMILY [illegible] WITH MINIMUM [illegible] MEMORY
ASR 33 (OR EQUIVALENT)
[illegible]
AT LEAST ONE OF THE FOLLOWING
[illegible]
[illegible]
[illegible] LINES, [illegible] LINES

[illegible]

[illegible]

3.2 [illegible]

[illegible]

3. [illegible]

3.1 [illegible]

[illegible]

[illegible]

[illegible] LOADER STARTING ADDRESS +D0

[illegible] + SIZE

[illegible]

3.1.1 PLACE ACCESS [illegible] SWITCH REGISTER.

3.1.2 DEPRESS 'LOAD ADDRESS' [illegible] AND RELEASE.

3.1.3 DEPRESS [illegible] ON CONSOLE AND RELEASE [illegible]

4. STARTING PROCEDURE

[illegible]

```
[illegible]
```

THE ABOVE IS ONLY AN EXAMPLE. THIS WOULD INDICATE THE SET OF
OPTIONS AT ADD. [illegible] IN THE PROGRAM. [illegible]
[illegible] BY THE USER. IF AUTO SIZING IS DONE. FOR INFORMATION ON SIZING
THIS SEE SECTION 6.4 FOR HELP.

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

4.1 CONTROL SWITCH SETTINGS

NOTE: IF THERE IS NO READ SWITCH(S), SW MAY BE MODIFIED AT LOC 176
OR BY HITTING CONTROL "G" "G" ON CONSOLE TERMINAL.

```
[illegible]  SET:  HALT ON ERROR
             SET:  LOOP ON CURRENT TEST
             SET:  INHIBIT ERROR PRINT OUT
             SET:  INHIBIT **ALL** TYPE OUT BELL ON ERROR.
             SET:  INHIBIT ITERATIONS.  [illegible] PASS
             SET:  ESCAPE TO NEXT TEST
             SET:  LOOP WITH CURRENT DATA
             SET:  CATCH ERROR AND LOOP ON IT
             SET:  USE PREVIOUS STATUS TABLE.  CLR-DO AUTO SIZE.
             SET:  RESERVED
             SET:  RESERVED
             SET:  RESERVED
             SET:  RESERVED
             SET:  LOOP ON SELECTED TEST
             SET:  RESTART PROGRAM AT SELECTED TEST
             SET:  RESELECT DEVICES DESIRED ABOVE.
```

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 [illegible]

SW 01 [illegible]

SW 09 [illegible]

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

1. SW 12 DELETE PRINT OUT BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 14 HALT ON THE ERROR.
4. SW 15 GOTO BEGINNING OF THE TEST ON ERROR.
5. SW 16 GOTO NEXT TEST ON ERROR.

SCOPE SWITCHES

1. SW 08 IF ENABLED BY 'SCOPE', ON AN ERROR, IF AN '*' IS PRINTED IN FRONT OF THE TEST NO. (EX. *TEST NO. 10) THEN SCOPE HAS INCORPORATED IN THAT TEST AND THEREFORE SW08 IS A SCOPE TEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW15=0, SW08=1, SW09=0). IF 'SCOPE' IS NOT ENABLED AND THERE IS A '*' BEFORE CONSTANT : SW08 IS BEST.
(SW14=1, SW15=0, SW09=0, SW08=1) FOR INTERMITTENT ERRORS: SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NO ERROR.
(SW14=1, SW15=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 DZQ11 DIAGNOSTICS PREVIOUSLY MENTIONED EXCEPT FOR ADDRESS 000242 FOR THE MODEM CONTROL AND CABLE TESTS AND 000310 FOR THE MODEL PARAMETER INPUT PROGRAM.

NOTE: IF ADDRESS 000242 IS NON-ZERO THE PROGRAM ASSUMES IT IS RUNNING UNDER ACT11 OR XXDP (CONTROL) AND WILL NOT AUTOSIZE. AFTER ALL AVAILABLE DZQ11'S ARE TESTED THE PROGRAM WILL REPORT IT WAS UNDER 'ACT-11'.

5. OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION 3.0 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 NATURE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT: LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE INTERPRETED.

6. ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 15=1). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE TYPE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.2 ERROR RECOVERY

IF FOR SOME REASON THE DV11 SHOULD "HANG THE BUS" (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN: LOOK IN LOCATION "TSTNO" (ADDRESS 1234) 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 HAS 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

DVII TRIAL PROGRAM MUST BE RUN PRIOR TO THE FIRST AND ONLY THE FIRST RUNNING OF ANY DVII DIAGNOSTIC IF "AUTO SIZING" IS NOT USED.
NOTE: IF NO PROGRAM OTHER THAN A DVII DIAGNOSTIC WAS LOADED AFTER DVII TRIAL OR IF CORE MEMORY HAS NOT BEEN CHANGED OR IF THERE IS NO DVII CONFIGURATION CHANGES; THE DVII TRIAL PROGRAM NEED NEVER BE RUN AGAIN. HOWEVER IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DVII 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 DVII 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 DVII'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 DZDVA-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 LENGTHY. THEREFORE EACH CHAR IS AN "END PASS AND THE ENTIRE "END" IS NOT REQUIRED FOR ACCEPTANCE.

9.4 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 SEQUENTIALY. THEY CONTAIN THE CSR,VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DV11.

D.ACTV (1276) EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DV11 WILL BE TESTED IN TURN. EXAMPLE: (DVACTV) 1276/0000000000C01111 MEANS THAT DV11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DVACTV) 1276/0000000000010001 MEANS THAT DV11 NO. 00,04 WILL BE TESTED.

DVSCR (1356) CONTAINS THE RECEIVER CSR OF THE CURRENT DV11 UNDER TEST.

LOO.33 (1412)
LO4.07 (1414)
LO8.11 (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 OF AN ASYNC LINE CARD.

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

```
                'MAP OF DV11 STATUS'
1500    175000
1502    000300
1504    000226
1506    000062
1510    000226
1512    000062
1514    004000
1516    000000
1520    004000
1522    000000
```

SYNC 'A' AND SYNC'B' MUST BE SET TO ZEROS FOR AN ASYNC LINE CARD.
THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 9 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

475 SPECIFIC CONFIGURATION.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

THE PROGRAM WILL START AT ADDRESS 175000 AND START 'REFERENCEING' ADDRESS. IF A NON-EX MEMORY TRAP OCCURES; THE POINTER (HOLDING 175000) IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTILL ADDRESS 175400 IS REACHED. IF A 'SLAVE SYNC RESPONSE' WAS ISSUED BY THE DV11 (OR ANY OTHER DEVICE) (NO NXM TRAP)(AND IT(SEL0)WAS=0) ; POINTER PLUS 12 (SEL12) IS TESTED TO CONTAIN 177777 (MUST BE EXACTLY 177777); IF A TRAP IS ENCOUNTERED OR IF SEL12 DOES NOT CONTAIN 177777 THE ABOVE UPDATING IS PERFORMED. IF SEL12 WAS EQUAL TO 177777 THE POINTER IS STORED AWAY AND THE ROUTINE CONTINUES AS ABOVE:
NOTE: IF THE PROGRAM DOES NOT FIND YOUR DV11; SOMETHING IS WRONG AND AUTO SIZING SHOULD NOT BE DONE.

8.5.2 FINDING THE VECTOR

THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND '.+2' (NEXT ADDRESS). BIT7 AND BIT6 (RX INTERUPT AND RX INTERUPT IE) ARE SET INTO DVSCR REGISTER; A DELAY IS MADE AND IF NO INTERUPT 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 INTERUPT OCCURED; THE ADDRESS TO WHICH THE DV11 INTERUPTED TO IS PICKED UP AND REPORTED AS THE VECTOR. NOTE: IF THE VECTOR REPORTED IS NOT THE VECTOR SET UP BY YOU; THERE IS A PROBLEM AND AUTO SIZING SHOULD NOT BE DONE.

8.5.3 PARAMETER ASSUMPTIONS.

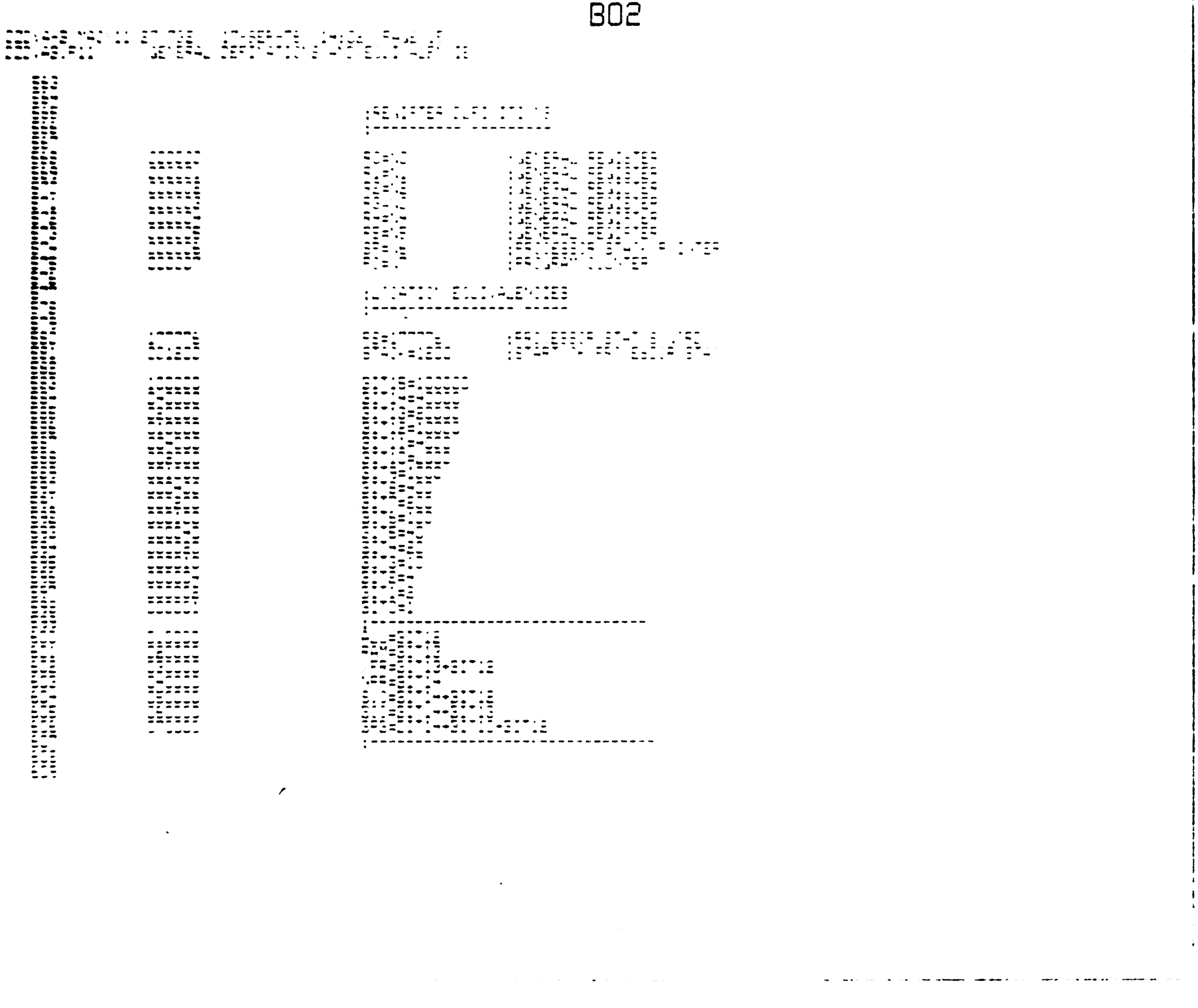
SINCE TOO MUCH HARDWARE WOULD NEED TO BE TURNED ON TO SIZE THE REST OF THE PARAMETERS; THE PROGRAM MUST ASSUME THE REMAINING VARIATIONS. THE RESULT IF NOT TO YOUR SPECIFIC CONFIGURATION MAY BE ALTERED BY HAND (TOGGLE IN) IS DESIRED. IN THIS WAY 95% OF THE PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.
THEREFORE:

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

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

NO1


```
                                ;*MAINDEC-11-DZDVA-B/(377) BASIC DV11 CONTROLLER MODULES TESTING
                                ;*COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                                ;*-------------------------------------------------------------

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "MAINDEC-11-DZDVA-B/(377)/BASIC DV11 CONTROLLER MODULES TESTI
                                        ;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                        SW15=100000             ;=1,HALT ON ERROR
  040000                        SW14=40000              ;=1,LOOP ON CURRENT TEST
  020000                        SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
  010000                        SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR.
  004000                        SW11=4000               ;=1,INHIBIT ITERATIONS
  002000                        SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
  001000                        SW09=1000               ;=1,LOOP WITH CURRENT DATA
  000400                        SW08=400                ;=1,LOOP ON ERROR
  000200                        SW07=200                ;=1, DO "AUTO SIZING" ON INITAL START UP.
  000100                        SW06=100
  000040                        SW05=40
  000020                        SW04=20
  000010                        SW03=10
  000004                        SW02=4                  ;LOCK ON TEST SELECT
  000002                        SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
  000001                        SW00=1                  ;RESELECT DV11 DESIRED ACTIVE
                                                        ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```



```
                        ;*****************************************************************
                        ;*****************************************************************
                        ;TRAP CATCHER FOR ILLEGAL INTERRUPTS
                        ;THIS ROUTINE "DATA TRAPPED" IF ENABLED
                        ;AND ANY UNEXPECTED INTERRUPT OCCURS.
                        ;IT WILL LOOP AT ".+2" FOR EASY DEBUG.
                        ;*****************************************************************
                        ;*****************************************************************

                .=0
                        ;STANDARD INTERRUPT VECTORS
                        ;-------------------------

                .=20
                        ;POWER FAIL VECTOR
                        ;NEW PC
                        ;NEW PS
                        ;EMT VECTOR
                        ;NEW PC
                        ;NEW PS
                .=40
                        ;ILLEGAL OR RESERVED INSTRUCTION TRAP VECTOR
                        ;NEW PC
                        ;NEW PS

                .=60
                        ;TTY KEYBOARD VECTOR
                        ;NEW PC
                        ;NEW PS
                .=200
                        ;START OF PROGRAM
000200  000137  000746          JMP     .START          ;GO TO START OF PROGRAM

                .=1200
001200  042516  047101  MSG:    .ASCII  /?? IS MAINDEC-11-D0A-A-B ??  BASIC I/O CONTROLLER MODULES TESTING
001220  000000          .WORD   0
001222  000000  SWR:    .WORD   0
                        ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                        ;-------------------------------------------------

001224  177560  TKSR:   .WORD   177560            ;TELETYPE KEYBOARD CONTROL REGISTER
001226  177562  TKBR:   .WORD   177562            ;TELETYPE KEYBOARD DATA BUFFER
001230  177564  TPSR:   .WORD   177564            ;TELEPRINTER CONTROL REGISTER
001232  177566  TPBR:   .WORD   177566            ;TELEPRINTER DATA BUFFER
                        ;PROGRAM CONTROL PARAMETERS
                        ;-------------------------

001234  000000  RETURN: .WORD   0                 ;SCOPE ADDRESS FOR LOOP ON TEST
001236  000000  LOOP:   .WORD   0                 ;ADDRESS OF NEXT TEST TO BE EXECUTED
                                                  ;ADDRESS FOR LOOP ON CURRENT TEST
```

D02

```
                                                ;PROGRAM CONTROL FLAGS
                                                ;--------------------

                                        .NIF==: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
                                        ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
                                        LOOPFLG:.BYTE   0               ;LOOP ON CURRENT TEST FLAG
                                        SVRFLG: .BYTE   0               ;SCOPE VERIFY FLAG.
                                                                        ;ON FIRST PASS OF EACH SVII ITERATIONS WILL BE SUPPRESSED
                                        .EVEN
                                        MTEST

                                                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                                ;POINTERS TO SUBROUTINES CAN BE FOUND
                                                ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS

                                        ;;**************************************************************************
                                        ;--------------------------------------------------------------------------
                                        ;TRPTAB:
                                        SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
                                                .SCOPE
                                        SCOPL=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
                                                .SCOPL
                                        TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
                                                .TYPE
                                        INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
                                                .INSTR
                                        INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
                                                .INSTER
                                        PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
                                                .PARAM
                                        SAVCS=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
                                                .SAVCS
                                        RESCS=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
                                                .RESCS
                                        CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
                                                .CONVRT
                                        CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR LF.
                                                .CNVRT
                                        MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
                                                .MSTCLR
                                        RAMCLR=TRAP+13          ;CALL TO CLEAR THE RAMS
                                                .RAMCLR
                                        DELA =TRAP+14           ;CALL TO VARIABLE DELAY COUNTER
                                                .DELAY
                                        ROMCLR=TRAP+15          ;CALL TO CLOCK ROM ONCE
                                                .ROMCLR
                                        DATACLR=TRAP+16         ;CALL TO CLR DATA
                                                .DATACLR

                                        ;--------------------------------------------------------------------------
                                        ;;**************************************************************************
```

```
                                        ;DV11 VECTOR AND REGISTER INDIRECT POINTERS

[redacted]  DVRVEC:  [redacted]          ;POINTER TO DV11 RECEIVER INTERRUPT VECTOR
[redacted]  DVRLVL:  [redacted]          ;POINTER TO DV11 RECEIVER INTERRUPT PRIORITY
[redacted]  DVTVEC:  [redacted]          ;POINTER TO DV11 TRANSMITTER INTERRUPT VECTOR
[redacted]  DVTLVL:  [redacted]          ;POINTER TO DV11 TRANSMITTER INTERRUPT PRIORITY
[redacted]  DVCSR:   [redacted]          ;POINTER TO DV11 SYSTEM CONTROL REGISTER
[redacted]  DVCSR1:  [redacted]          ;POINTER TO DV11 SYSTEM CONTROL REGISTER #1.
[redacted]  DVRBUF:  [redacted]          ;POINTER TO DV11 NEXT RECEIVED CHARACTER BUFFER
[redacted]  DVLPR:   [redacted]          ;POINTER TO DV11 LINE PARAMETER REGISTER
[redacted]  DVSR2:   [redacted]          ;POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER
[redacted]  DVSR3:   [redacted]          ;POINTER TO DV11 SECONDARY REGISTER #3.
[redacted]  DVSR4:   [redacted]          ;POINTER TO DV11 SECONDARY REGISTER #4
[redacted]  DVSR5:   [redacted]          ;POINTER TO DV11 SPECIAL FUNCTIONS REGISTER
[redacted]  DVNXR:   [redacted]          ;POINTER TO DV11 NXM STATUS REGISTER
[redacted]  RESVEC:  [redacted]          ;POINTER TO RESERVED REGISTER.

                                        ;DV11 CONTROL INDICATORS FOR CURRENT DV11 UNDER TEST
                                        ;--------------------------------------------------

[redacted]  MASK.0:  .BYTE [redacted]    ;LAST CHAR TO TEST AND PARITY MASK FOR LINE 00
[redacted]  MASK.1:  .BYTE [redacted]    ;LAST CHAR TO TEST AND PARITY MASK FOR LINE 01
[redacted]  MASK.2:  .BYTE [redacted]    ;LAST CHAR TO TEST AND PARITY MASK FOR LINE 02
[redacted]  MASK.3:  .BYTE [redacted]    ;LAST CHAR TO TEST AND PARITY MASK FOR LINE 03

[redacted]  CLK.0:   .BYTE 0.            ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03
[redacted]  CLK.4:   .BYTE 0.            ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07
[redacted]  CLK.8:   .BYTE 0.            ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11
[redacted]  CLK.C:   .BYTE 0.            ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 12-15

[redacted]  LPR.0:   [redacted]          ;PARAMETERS FOR LINE 00
[redacted]  LPR.4:   [redacted]          ;PARAMETERS FOR LINE 04
[redacted]  LPR.8:   [redacted]          ;PARAMETERS FOR LINE 08
[redacted]  LPR.12:  [redacted]          ;PARAMETERS FOR LINE 12

[redacted]  SYNC0A:  [redacted]          ;SYNC 0
[redacted]  SYNC1A:  [redacted]          ;
[redacted]  SYNC0B:  [redacted]          ;
[redacted]  SYNC1B:  [redacted]          ;

                                        ;SUMMARY
                                        ;-------

;       MASK.Y                  [redacted]      ;[redacted] BITS PER CHAR.
;                               [redacted]      ;[redacted] BITS PER CHAR.
;                               [redacted]      ;[redacted] BITS PER CHAR.

;       CLK.X                   [redacted]      ;[redacted] BITS PER CHAR.
;                               [redacted]      ;[redacted] BITS PER CHAR.
;                               [redacted]      ;[redacted] BITS PER CHAR.
```

```
                                        ;DV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                        ;-----------------------------------------

                        000530                  .=1500
                                        DVTABL:
000000 000000           000530          DVCR00: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 00
000001 000000           000531          DVTR00: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 00
000002 000000           000532          DV00.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
000003 000000           000533          SYNA00: .BLKW 1         ;SYNC TWO
000004 000000           000534          DV00.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
000005 000000           000535          SYNB00: .BLKW 1         ;SYNC TWO
000006 000000           000536          DV00.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
000007 000000           000537          SYNC00: .BLKW 1         ;SYNC TWO
000010 000000           000540          DV00.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
000011 000000           000541          SYND00: .BLKW 1         ;SYNC TWO

000012 000000           000542          DVCR01: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 01
000013 000000           000543          DVTR01: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 01
000014 000000           000544          DV01.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
000015 000000           000545          SYNA01: .BLKW 1         ;SYNC TWO
000016 000000           000546          DV01.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
000017 000000           000547          SYNB01: .BLKW 1         ;SYNC TWO
000020 000000           000550          DV01.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
000021 000000           000551          SYNC01: .BLKW 1         ;SYNC TWO
000022 000000           000552          DV01.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
000023 000000           000553          SYND01: .BLKW 1         ;SYNC TWO

000024 000000           000554          DVCR02: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 02
000025 000000           000555          DVTR02: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 02
000026 000000           000556          DV02.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
000027 000000           000557          SYNA02: .BLKW 1         ;SYNC TWO
000030 000000           000560          DV02.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
000031 000000           000561          SYNB02: .BLKW 1         ;SYNC TWO
000032 000000           000562          DV02.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
000033 000000           000563          SYNC02: .BLKW 1         ;SYNC TWO
000034 000000           000564          DV02.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
000035 000000           000565          SYND02: .BLKW 1         ;SYNC TWO

000036 000000           000566          DVCR03: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 03
000037 000000           000567          DVTR03: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 03
000040 000000           000570          DV03.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
000041 000000           000571          SYNA03: .BLKW 1         ;SYNC TWO
000042 000000           000572          DV03.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
000043 000000           000573          SYNB03: .BLKW 1         ;SYNC TWO
000044 000000           000574          DV03.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
000045 000000           000575          SYNC03: .BLKW 1         ;SYNC TWO
000046 000000           000576          DV03.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 03
000047 000000           000577          SYND03: .BLKW 1         ;SYNC TWO

000050 000000           000600          DVCR04: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 04
000051 000000           000601          DVTR04: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 04
000052 000000           000602          DV04.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
000053 000000           000603          SYNA04: .BLKW 1         ;SYNC TWO
000054 000000           000604          DV04.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
000055 000000           000605          SYNB04: .BLKW 1         ;SYNC TWO
000056 000000           000606          DV04.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04
```

```
                        SYNC04: .BLKW 1         ;SYNC TWO
                        DV04.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
                        SYND04: .BLKW 1         ;SYNC TWO

                        DVCR05: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 05
                        DVTR05: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 05
                        DV05.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
                        SYNA05: .BLKW 1         ;SYNC TWO
                        DV05.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
                        SYNB05: .BLKW 1         ;SYNC TWO
                        DV05.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
                        SYNC05: .BLKW 1         ;SYNC TWO
                        DV05.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
                        SYND05: .BLKW 1         ;SYNC TWO

                        DVCR06: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 06
                        DVTR06: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 06
                        DV06.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
                        SYNA06: .BLKW 1         ;SYNC TWO
                        DV06.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
                        SYNB06: .BLKW 1         ;SYNC TWO
                        DV06.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
                        SYNC06: .BLKW 1         ;SYNC TWO
                        DV06.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
                        SYND06: .BLKW 1         ;SYNC TWO

                        DVCR07: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 07
                        DVTR07: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 07
                        DV07.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
                        SYNA07: .BLKW 1         ;SYNC TWO
                        DV07.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
                        SYNB07: .BLKW 1         ;SYNC TWO
                        DV07.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
                        SYNC07: .BLKW 1         ;SYNC TWO
                        DV07.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 07
                        SYND07: .BLKW 1         ;SYNC TWO

                        D..END: 000000
```

```
                                        ;PROGRAM INITIALIZATION
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;CLEARS PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

001000  012737  177776  000340  .START: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
001006  012706  001000                  MOV     #STACK,SP               ;SET UP STACK
001012  012737  000114  000340          MOV     #.BFAIL,@#4             ;SET UP POWER FAIL ERROR
001020  013746  001334                  MOV     @DVNUM,SAVNUM           ;SAVE NUMBER OF DEVICES IN SYSTEM.
001024  005037  001330                  CLR     PASCNT                  ;CLEAR PASS COUNT
001030  005037  001332                  CLR     ERRFLG                  ;CLEAR ERROR FLAG
001034  005037  001312                  CLR     QV.FLG                  ;ZERO QUICK VERIFY FLAG
001040  012737  001306  001304          MOV     #QV.MAP,CREAM           ;GET MAP POINTER.
001046  113737  001304  001316          MOVB    @#1,RUN                 ;POINT POINTER TO FIRST DEVICE.
001054  005037  001336                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
001060  005037  001340                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
001064  012737  000001  001314          MOV     #1,TSTNO                ;SET UP FOR TEST 1
001072  012737  001142  000114          MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
001100  105737  001310                  TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
001104  001063                          BNE     1$                      ;BR IF YES
001106  013746  000004                  MOV     4,-(SP)                 ;
001112  005037  000004                  CLR     4                       ;
001116  005037  177560                  CLR     @#60                    ;
001122  012737  000104  000004          MOV     #BOOT,4                 ;
001130  005777  177102                  TST     @SWR                    ;
001134  000240                          NOP                             ;
001136  000407                          BR      81$                     ;
001140  022626                  80$:    CMP     (SP)+,(SP)+             ;
001142  012737  000174  001300          MOV     #LIGHT,LIGHTS           ;
001150  010737  000176  001302          MOV     R0,@SWR.SWR             ;
001156  012637  000006                  MOV     (SP)+,6                 ;
001162  012637  000004          81$:    MOV     (SP)+,4                 ;
001166  104402  001000                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
001172  105137  001310                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
001176  105777  177034                  TSTB    @SWR                    ;BIT7=1?
001202  100402                          BMI     16$                     ;BR IF NO AUTO SIZE
001204  004737  006634                  JSR     PC,CSRMAP               ;DO THE AUTO SIZE
001210  104402  005461          16$:    TYPE    ,XHEAD          ;TYPE HEADER
001214  012737  001500  001246          MOV     #QV.MAP,TEMP1           ;SET POINTER
001222  017737  177554  001250  5$:     MOV     @TEMP1,TEMP2            ;SET DATA
001230  022737  177777  001250          CMP     #177777,TEMP2           ;ALL DONE?
001236  001406                          BEQ     1$                      ;BR IF YES
001240  104410                          CONVRT                          ;
001242  005506                          XSTATG                          ;
001244  062737  000002  001246          ADD     #2,TEMP1                ;UPDATE POINTER
001252  000763                          BR      5$                      ;
001254  005737  000042          1$:     TST     @#42                    ;IS PROGRAM RUNNING UNDER MONITOR
001260  001430                          BNE     3$                      ;BR IF YES
001262  032777  000001  176746          BIT     #SW00,@SWR              ;SELECT SPECIFIC DEVICES``
001270  001424                          BEQ     3$                      ;BR IF NO.
001272  104402  005432                  TYPE    ,MNEW                   ;TYPE THE MESSAGE.
001276  005037                          CLR     R0                      ;ZERO DATA LIGHTS
```

```
                                                HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
                                                CMPB    @#SWR.SAVACT    ;IS THE NUMBER VALID?
                                                BLOS    2$              ;BR IF NUMBER IS OK.
                                                TYPE    .MERR3          ;TELL USER OF INVALID NUMBER.
                                                HALT                    ;STOP EVERY THING.
                                                BR      .-2             ;RESTART THE PROGRAM AGAIN.
                                        1$:     MOVB    @#SWR,DVACTV    ;GET NEW DEVICE PATTERN
                                                MOVB    DVACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
                                                BIC     #^C<377>,R0     ;USE ONLY LOW BYTE.
                                                HALT                    ;CONTINUE DYNAMIC SWITCHES.
                                        3$:     MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
                                                MOV     #302,R1         ;VECTOR AREA. 300-776
                                        4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
                                                CLR     (R1)+           ;IN VECTOR AREA.
                                                CMP     (R0)+,(R1)+     ;POP POINTERS
                                                CMP     #1000,R0        ;ALL DONE??
                                                BNE     4$              ;BR IF NO.

                                                ;TEST START AND RESTART
                                                ;----------------------

                                .BEGIN: MOV     #340,PS         ;LOCK OUT INTERRUPTS
                                                MOV     #STACK,SP       ;SET UP STACK
                                                TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
                                                BNE     3$              ;BR IF YES
                                                BIT     #BIT2,@#SWR     ;CHECK FOR LOCK ON TEST
                                                BEQ     1$              ;BR IF NO LOCK DESIRED.
                                                TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
                                                MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
                                                MOV     #NOP,TTST+2     ;SET UP TO LOCK
                                                BR      2$              ;CONTINUE ALONG.
                                        1$:     MOV     @#RSW,TTST      ;PREPARE NORMAL SCOPE ROUTINE
                                                MOV     @#RSX,TTST+2    ;LOCK NOT SELECTED. SET UP FOR NORMAL SCOPE LOOP
                                        2$:
                                        3$:     MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
                                        4$:     TYPE    ,MR             ;TYPE R
                                                JMP     @RETURN         ;START TESTING
```

```
                                                        ;END OF PASS
                                                        ;TYPE NAME OF TEST
                                                        ;UPDATE PASS COUNT
                                                        ;CHECK FOR EXIT TO ACT-11
                                                        ;RESTART TEST

 002436  000005                       .EOP:   RESET                   ;MAKE THE  WORLD CLEAN AGAIN.
 002440  005037  001234                       CLR     LSTERR          ;CLEAR LAST ERROR PC
 002444  105037  001311                       CLRB    ERRFLG          ;CLEAR ERROR FLAG
 002450  005237  001230                       INC     PASCNT          ;UPDATE PASS COUNT
 002454  013777  001230  176516               MOV     PASCNT,@LIGHTS  ;DISPLAY PASS COUNT
 002462  104402  005145                       TYPE    .MEPASS         ;TYPE END PASS
 002466  104402  005330                       TYPE    .MCSRX          ;TYPE CSR
 002472  104411  002604                       CNVRT   .XCSR           ;SHOW IT
 002476  104402  005336                       TYPE    .MVECX          ;TYPE VECTOR
 002502  104411  002612                       CNVRT   .XVEC           ;SHOW IT
 002506  104402  005344                       TYPE    .MPASSX         ;TYPE PASSES
 002512  104411  002620                       CNVRT   .XPASS          ;SHOW IT
 002516  104402  005355                       TYPE    .MERRX          ;TYPE ERRORS
 002522  104411  002626                       CNVRT   .XERR           ;SHOW IT
 002526  105337  001303                       DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
 002532  001017                               BNE     RESTRT          ;BR IF NO.
 002534  112737  000377  001313               MOVB    #377,QV.FLG     ;SET THE QUICK VERIFY FLAG.
 002542  113737  001301  001303               MOVB    DVNUM,SAVNUM    ;RESTORE THE COUNT
 002550  013701  000042                       MOV     @#42,R1         ;CHECK FOR ACT-11 OR DDP
 002554  001406                               BEQ     RESTRT          ;IF NOT, CONTINUE TESTING
 002556  000005                               RESET                   ;STOP THE SHOW--CLEAR THE WORLD
 002560                              LOGICAL:
 002560  004711                               JSR     PC,(R1)
 002562  000240                               NOP
 002564  000240                               NOP
 002566  000240                               NOP
 002570  000240                               NOP
 002572  012737  005666  001214  RESTRT:     MOV     #CYCLE,RETURN
 002600  000137  005666                       JMP     CYCLE
 002604  000001                      XCSR:    1
 002606     006     002                       .BYTE   6,2
 002610  001362                               DVSCR
 002612  000001                      XVEC:    1
 002614     003     002                       .BYTE   3,2
 002616  001352                               DVRVEC
 002620  000001                      XPASS:   1
 002622     006     002                       .BYTE   6,2
 002624  001230                               PASCNT
 002626  000001                      XERR:    1
 002630     006     002                       .BYTE   6,2
 002632  001232                               ERRCNT

                                              ;SCOPE LOOP AND INTERATION HANDLER
                                              ;--------------------------------

 002634                              .SCOPE:
 002634  022737  177570  001202               CMP     #177570,SWR     ;IS THERE A REAL SWR?
 002642  001411                               BEQ     64$             ;BR IF YES
 002644  017746  176336                       MOV     @TKDBR,-(SP)    ;SAVE KEYBOARD CHAR
 002650  042716  000200                       BIC     #BIT7,(SP)      ;CLEAR PARITY BIT
```

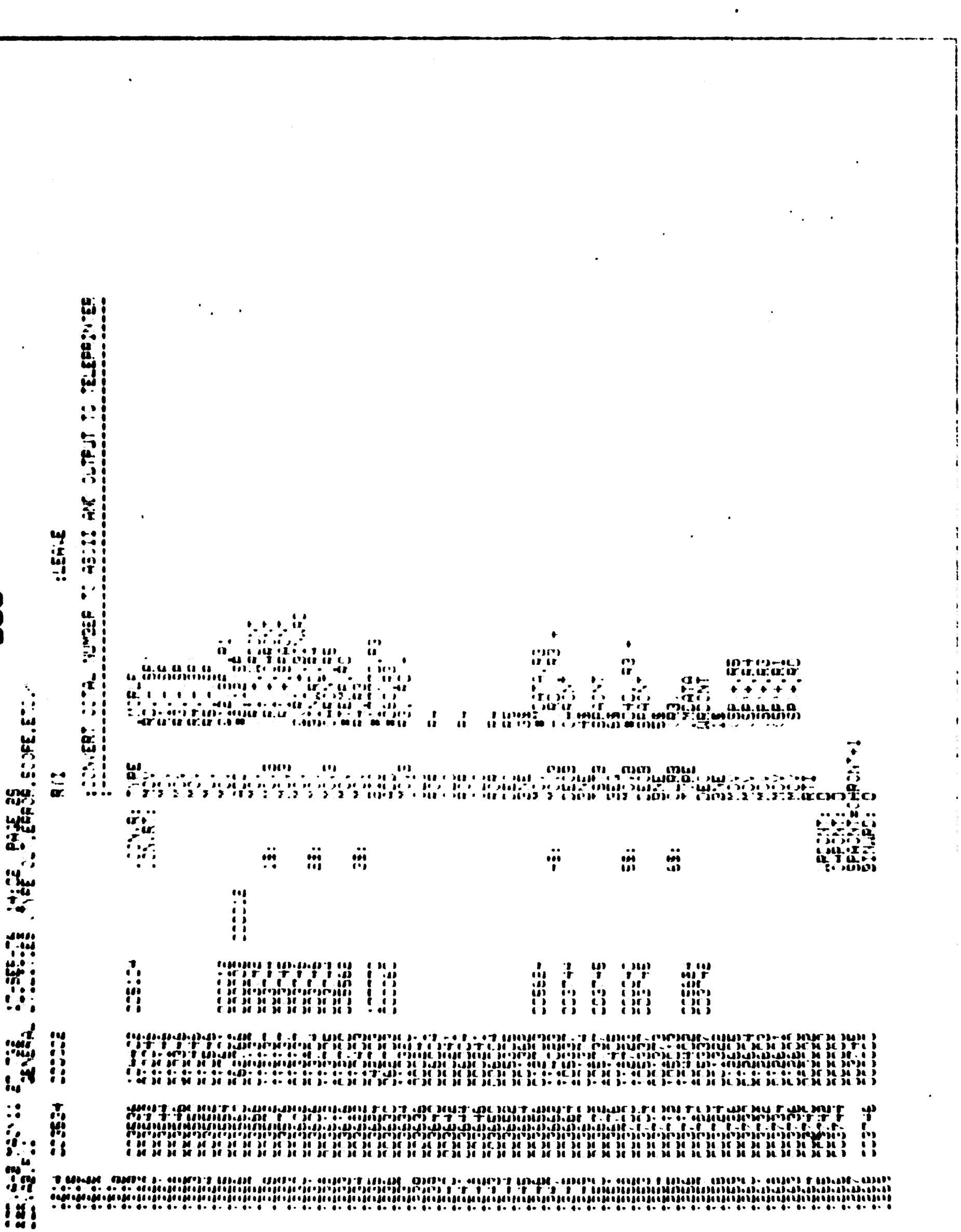
```
DZDVA-B MACY11 27.732  17-SEP-76  14:02  PAGE 35
DZDVAB.P11     GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

 1046  003654  122726  000007                  CMPB    #7,(SP)+        ;WAS IT CNTRL 'G' ?
 1047  003660  001002                          BNE     .+6             ;BR IF NO.
 1048  003662  004737  004640                  JSR     PC,SERV.G       ;SERVICE "CNTRL 'G'".
 1049  003666  005037  001234          64$:    CLR     LSTERR          ;CLEAR LAST ERROR PC.
 1050  003672  010016                          MOV     R0,(SP)         ;SAVE R0 ON THE STACK
 1051  003674  032777  040000  176300          BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
 1052  003702  001407                  TTST:   BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1: THIS LOC =240)
 1053  003704  000437                          BR      3$              ;GOTO 3$    (IF LOCK SW01=1: THIS LOC =240)
 1054  003706  105777  176272                  TSTB    @TKCSR          ;KEYBOARD DONE?
 1055  003712  100034                          BPL     3$              ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
 1056  003714  017700  176266                  MOV     @TKDBR,R0       ;CLEAR DONE BIT
 1057  003720  200415                          BR      2$              ;CONTINUE
 1058  003722  032777  004000  176252  1$:     BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
 1059  003730  001011                          BNE     2$              ;BR IF YES
 1060  003732  105737  001313                  TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
 1061  003736  001406                          BEQ     2$              ;BR IF QUICK PASS.
 1062  003740  005237  001224                  INC     LPCNT           ;UPDATE ITERATION COUNTER
 1063  003744  023737  001224  001222          CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
 1064  003752  001014                          BNE     3$              ;BR IF NOT YET
 1065  003754  105037  001311          2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
 1066  003760  005037  001224                  CLR     LPCNT           ;START ICOUNTER AT 0
 1067  003764  005037  001220                  CLR     LOCK
 1068  003770  012737  000020  001222          MOV     #20,ICOUNT      ;RESET ITERATIONS
 1069  003776  013737  001216  001214          MOV     NEXT,RETURN     ;GET NEXT TEST
 1070  004004  011600                  3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
 1071  004006  022626                          POP2SP                  ;FAKE AN "RTI"
 1072  004010  000177  176200                  JMP     @RETURN         ;GO DO THE TEST
 1073  004014  001407                  BRW:    1407
 1074  004016  000437                  SRX:    437
 1075
 1076                                          ;CHECK FOR FREEZE ON CURRENT DATA
 1077                                          ;-------------------------------
 1078
 1079  004020  032777  001000  176154  .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
 1080  004026  001405                          BEQ     1$              ;BR IF NOT SET.
 1081  004030  005737  001220                  TST     LOCK
 1082  004034  001402                          BEQ     1$
 1083  004036  013716  001220                  MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
 1084  004042  000002                  1$:     RTI                     ;GO BACK.
 1085
 1086                                          ;TELETYPE OUTPUT ROUTINE
 1087                                          ;----------------------
 1088
 1089  004044  010546                  .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
 1090  004046  017605  000002                  MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
 1091  004052  062766  000002  000002          ADD     #2,2(SP)        ;POP OVER ADDRESS.
 1092  004060  032777  010000  176114  1$:     BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT?
 1093  004066  001012                          BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
 1094  004070  105715                          TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1 BIT)?
 1095  004072  100002                          BPL     2$              ;BR IF NUMBER IS PLUS
 1096  004074  104402  005104                  TYPE    ,MCRLF          ;TYPE A CR/LF!
 1097  004100  105777  176104          2$:     TSTB    @TPCSR          ;TTY READY?
 1098  004104  100375                          BPL     2$              ;BR IF NO.
 1099  004106  112577  176100                  MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
 1100  004112  001362                          BNE     1$              ;IF NOT ZERO KEEP PRINTING!
 1101  004114  012605                  3$:     MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
```

```
1132  003116  000002                          RTI                     ;GO HOME
1133                                          ;-----------------------
1134
1135  003120  010346                .INSTR:   MOV     R3,-(SP)        ;SAVE R3 ON STACK
1136  003122  010446                          MOV     R4,-(SP)        ;SAVE R4 ON STACK
1137  003124  C17637  000004  003142          MOV     @4(SP),.MSG
1138  003132  062766  000002  000004          ADD     #2,4(SP)
1139  003140  104402                .INST1:   TYPE
1140  003142  000000                .MSG:     0
1141  003144  012704  005520                  MOV     #INBUF,R4
1142  003150  012703  000007                  MOV     #7,R3
1143  003154  105777  176024          1$:     TSTB    @TKCSR
1144  003160  100375                          BPL     1$
1145  003162  117714  176020                  MOVB    @TKDBR,(R4)
1146  003166  142714  000200                  BICB    #200,(R4)
1147  003172  122427  000015                  CMPB    (R4)+,#15
1148  003176  001417                          BEQ     INSTR2
1149  003200  105777  176004          2$:     TSTB    @TPCSR
1150  003204  100375                          BPL     2$
1151  003206  017777  175774  175776          MOV     @TKDBR,@TPDBR
1152  003214  005303                          DEC     R3
1153  003216  001356                          BNE     1$
1154  003220  012604                          MOV     (SP)+,R4
1155  003222  012603                          MOV     (SP)+,R3
1156  003224  104402  005100          .INSTE: TYPE    .MQM
1157  003230  010346                          MOV     R3,-(SP)
1158  003232  010446                          MOV     R4,-(SP)
1159  003234  000741                          BR      .INST1
1160  003236  012604                INSTR2:   MOV     (SP)+,R4        ;RESTORE R4
1161  003240  012603                          MOV     (SP)+,R3        ;RESTORE R3
1162  003242  000002                          RTI
1163
1164                                          ;CONVERT ASCII STRING TO OCTAL
1165                                          ;----------------------------
1166
1167  003244  010546                .PARAM:   MOV     R5,-(SP)
1168  003246  010446                          MOV     R4,-(SP)
1169  003250  016605  000004                  MOV     4(SP),R5
1170  003254  012537  003434                  MOV     (R5)+,LOLIM
1171  003260  012537  003436                  MOV     (R5)+,HILIM
1172  003264  012537  003440                  MOV     (R5)+,DEVADR
1173  003270  112537  003442                  MOVB    (R5)+,LOBITS
1174  003274  112537  003443                  MOVB    (R5)+,ADRCNT
1175  003300  010566  000004                  MOV     R5,4(SP)
1176  003304  005005                PARAM1:   CLR     R5
1177  003306  012704  005520                  MOV     #INBUF,R4
1178  003312  122714  000015                  CMPB    #15,(R4)
1179  003316  001420                          BEQ     PARERR
1180  003320  121427  000060          1$:     CMPB    (R4),#60
1181  003324  002415                          BLT     PARERR
1182  003326  121427  000067                  CMPB    (R4),#67
1183  003332  003012                          BGT     PARERR
1184  003334  142714  000060                  BICB    #60,(R4)
1185  003340  152405                          BISB    (R4)+,R5
1186  003342  122714  000015                  CMPB    #15,(R4)
1187  003346  001406                          BEQ     LIMITS
```

```
1158  003350  006305                          ASL     R5
1159  003352  006305                          ASL     R5
1160  003354  006305                          ASL     R5
1161  003356  000760                          BR      1$
1162  003360  104404                  PARERR: INSTER
1163  003362  000750                          BR      PARAM1
1164
1165                                          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1166                                          ;--------------------------------------
1167
1168  003364  020537  003436          LIMITS: CMP     R5,HILIM
1169  003370  101373                          BHI     PARERR
1170  003372  020537  003434                  CMP     R5,LOLIM
1171  003376  103770                          BLO     PARERR
1172  003400  133705  003442                  BITB    LOBITS,R5
1173  003404  001365                          BNE     PARERR
1174
1175                                          ;STORE NUMBER AT SPECIFIED ADDRESS
1176
1177  003406  013704  003440                  MOV     DEVADR,R4
1178  003412  010524                  1$:     MOV     R5,(R4)+
1179  003414  062705  000002                  ADD     #2,R5
1180  003420  105337  003443                  DECB    ADRCNT
1181  003424  001372                          BNE     1$
1182  003426  012604                          MOV     (SP)+,R4
1183  003430  012605                          MOV     (SP)+,R5
1184  003432  000002                          RTI
1185  003434  000000                  LOLIM:  0
1186  003436  000000                  HILIM:  0
1187  003440  000000                  DEVADR: 0
1188  003442  000000                  LOBITS: 0
1189          003443                  ADRCNT=LOBITS+1
1190
1191                                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
1192                                          ;------------------------------------
1193
1194  003444  016637  000004  001276  .SAVOS: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1195
1196                                          ;SAVE R0-R5.
1197
1198  003452  010537  001272          SVOS:   MOV     R5,SAVR5        ;SAVE R5
1199  003456  010437  001270                  MOV     R4,SAVR4        ;SAVE R4
1200  003462  010337  001266                  MOV     R3,SAVR3        ;SAVE R3
1201  003466  010237  001264                  MOV     R2,SAVR2        ;SAVE R2
1202  003472  010137  001262                  MOV     R1,SAVR1        ;SAVE R1
1203  003476  010037  001260                  MOV     R0,SAVR0        ;SAVE R0
1204  003502  000002                          RTI                     ;LEAVE.
1205
1206                                          ;RESTORE R0-R5
1207
1208  003504  013700  001260          .RESCS: MOV     SAVR0,R0        ;RESTORE R0
1209  003510  013701  001262                  MOV     SAVR1,R1        ;RESTORE R1
1210  003514  013702  001264                  MOV     SAVR2,R2        ;RESTORE R2
1211  003520  013703  001266                  MOV     SAVR3,R3        ;RESTORE R3
1212  003524  013704  001270                  MOV     SAVR4,R4        ;RESTORE R4
1213  003530  013705  001272                  MOV     SAVR5,R5        ;RESTORE R5
```

```
:LEAVE
```

```
:DEF P: ASSIGN ANY OUTPUT TO TELEPRINTER
```



C03


```
                                        ;TRAP DISPATCH SERVICE
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO [illegible] POINTER
                                        ;TO SELECTED SUBROUTINE

                        .TRPSR: [illegible]  [illegible] SP)      ;GET PC OF RETURN
        [illegible]             SUB     #2,(SP)                   ;[illegible] OF TRAP
                                [illegible] [illegible],SP        ;GET TRAP
                        TRPCN:  [illegible] [illegible]           ;MULTIPLY TRAP ARG BY 2
                                [illegible] #[illegible],SP       ;CLEAR UNWANTED BITS
                                [illegible] #.TRPTAB,SP           ;POINTER TO SUBROUTINE ADDRESS
                                [illegible] [illegible],SP        ;SUBROUTINE ADDRESS
                                JMP     @(SP)+                    ;GO TO SUBROUTINE

                                        ;ERROR HANDLER
                                        ;-------------

                        .HLT:
177570  001202                  CMP     #177570,SWR               ;IS THERE A REAL SWR
                                BEQ     64$                       ;BR IF YES
        [illegible]             MOV     @TKDBR,-(SP)              ;SAVE KEYBOARD CHAR
                                [illegible] #[illegible],(SP)     ;CLEAR PARITY BIT
                                CMPB    #[illegible],(SP)+        ;WAS IT CNTRL "G"
                                BNE     .+6                       ;[illegible] IF NO.
        [illegible]             [illegible] PC,SERV.G             ;SERVICE CNTRL "G".
        175140  64$:            [illegible] #SW12,@SWR            ;BELL ON ERROR
                                BEQ     XBX                       ;BR IF NO BELL
        175140                  [illegible]B @TPCSR               ;TTY READY.
                                [illegible] XBX                   ;DON'T WAIT IF TTY NOT READY.
        [illegible]             MOVB    #BOT,@TPCBF               ;PUSH A BELL AT THE TTY.
                        XBX:    [illegible] #SW13,@SWR            ;DELETE ERROR PRINT OUT
                                BNE     [illegible]               ;BR IF NO PRINT [illegible]
        001234                  CMP     (SP),LSTERR               ;WAS THIS ERROR FOUND LAST TIME?
                                BEQ     1$                        ;BR IF YES
        001234                  MOV     (SP),LSTERR               ;RECORD BEING HERE
        [illegible]             [illegible]B ERRFLG               ;PREPARE HEADER
                        1$:     [illegible] [illegible]           ;SAVE ALL PROC REGISTERS
                                MOV     (SP),R5                   ;GET THE PC OF ERROR
        000002                  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
        177001                  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 HEADER
                                BEQ     TYPMSG                    ;BR IF YES
        004276                  TST     DATABP                    ;DOES DATA TABLE EXIST?
                                BNE     TYPDAT                    ;BR IF YES.
                        TYPMSG: TYPE    .MCRLF
                                TYPE    .MCRLF
                                TST     LOOP
```

D03

```
                          GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)            PAGE 30

                                                BEQ     1$
                                                TYPE    ,MASTEK
                                        1$:     TYPE    ,MTSTN
                                                CNVRT   ,XTSTN          ;SHOW IT
                                                TYPE    ,MERRPC         ;TYPE PC.
                                                CNVRT   ,ERTABO         ;SHOW IT
                                                TYPE    ,MCRLF          ;GIVE A CR LF
                        001311                  MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
                004262                          TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
                                                BEQ     WRNO.EM         ;BR IF NO.
                                                TYPE                    ;TYPE
                                        ERRMSG: 0                       ;       ERROR MESSAGE
                                        WRNO.EM:
                004264                          TST     DATAHD          ;DATA HEADER?
                                                BEQ     TYPDAT          ;BR IF NO
                                                TYPE                    ;TYPE
                                        DATAHD: 0                       ;       DATA HEADER
                004276                  TYPDAT: TST     DATABP          ;DATA TABLE?
                                                BEQ     RESREG          ;BR IF NO.
                                                CONVRT                  ;SHOW
                                        DATABP: 0                       ;       DATA TABLE
                                        RESREG: RESREG                  ;RESTORE PROC REGISTERS
                174674                  HALTS:  TST     DSWR            ;HALT ON ERROR?
                                                BPL     EXITER          ;BR IF NO HALT ON ERROR
                                                PUSHR0                  ;SAVE R0
                000002                          MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
                                                HALT                    ;HALT
                                                POPR0                   ;GET R0
                001232                  EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
                                                BIT     #SW09,DSWR      ;GOTO TOP OF TEST?
                        174646                  BNE     1$              ;BR IF YES
                002000  174636                  BIT     #SW10,DSWR      ;GOTO NEXT TEST?
                                                BEQ     2$              ;BR IF NO
                001616  001214                  MOV     NEXT,RETURN     ;SET FOR NEXT TEST
                                        1$:     MOV     #STACK,SP       ;RESET SP
                                                JMP     @RETURN         ;GO TO SPECIFIED TEST
                                        2$:     RTI                     ;RETURN
                                        ERTABO:
                002                             .BYTE   6,2
                                                SAVPC
                                        XTSTN:
                002                             .BYTE   3,2
                                                TSTNO
                                                ;ENTER HERE ON POWER FAILURE
                                                ;---------------------------

        004400                          .PFAIL:
        004400  012737  004414  000024          MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
                000000                          HALT                            ;HALT ON POWER DOWN NORMAL
                000777                          BR      .

                                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

        004414                          RESTAR:
        004414  012737  004402  000024          MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
```

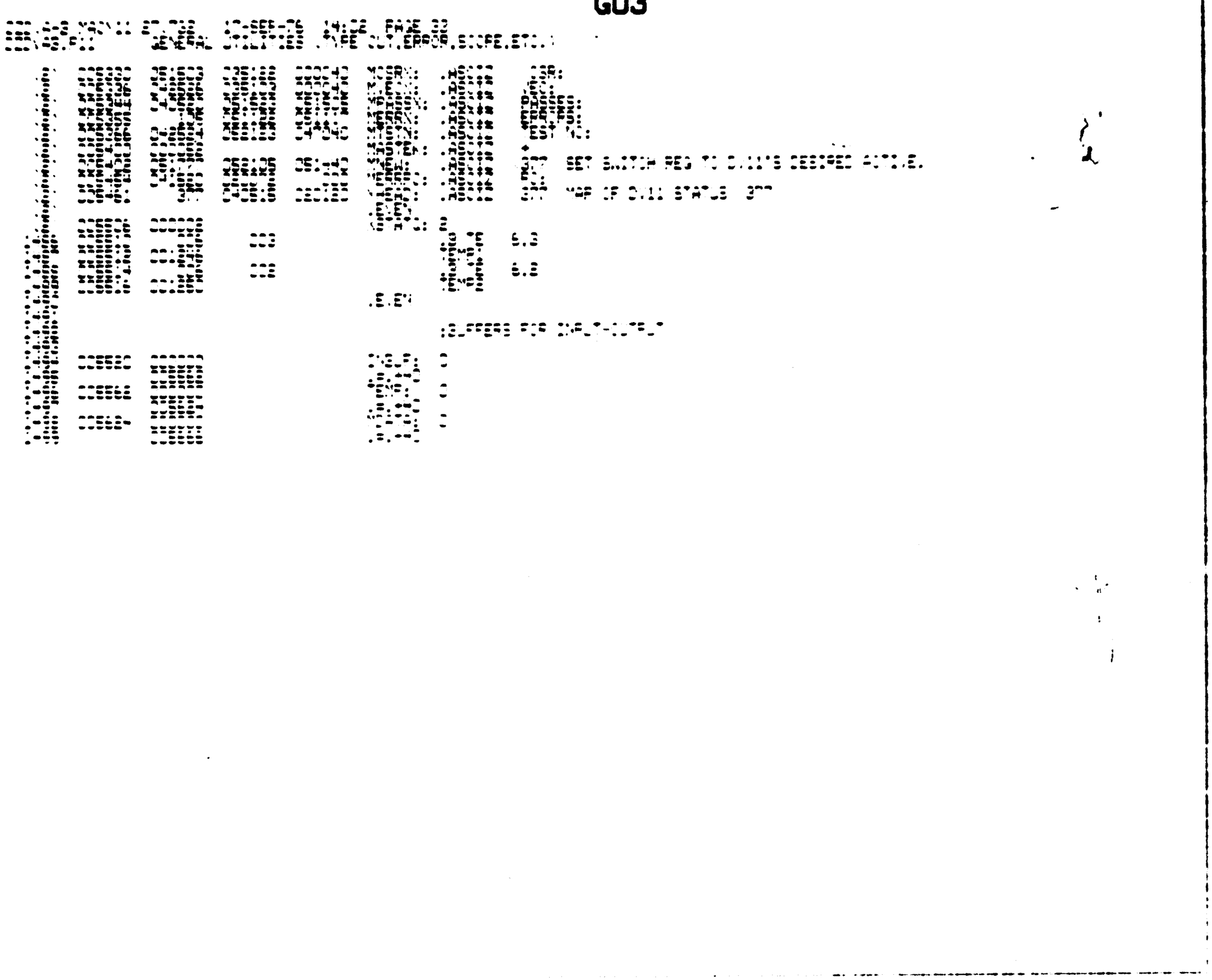
E03

```
                                .ROMCLR:
                                        MOV     @#RESET.SD.SIR  ;ISSUE MASTER CLEAR.
                                        RTI

                                .ROMCLR:
                                        BIS     @#BIT1.SDVSIR
                                        RTI

                                .DATFCLK:
                                        MOV     R0,-(SP)
                                        CLR     R0
                                        BIS     @#BIT8.SDVLCR
                                        MOV     #DVLCR.SS
                                        RORB    R0
                                        BCC     .+4
                                        INC     R0
                                        SWAB    R0
                                        BIS     #1,R0
                                10$:    MOV     R0,@#DVLCR
                                        MOV     (SP)+,R0
                                        RTI

                                .BLKW 1
```

F03


```
[illegible]  004000  174336  SER..G: BIT    #4000,@TKCSR    ;BY [illegible]
[illegible]                          [illegible] SER.4       ;[illegible]
[illegible]  174326  [illegible]     MOV    @SWR,90$        ;SAVE SWR .
[illegible]  174316  1$:             MOV    90$,@SWR        ;
[illegible]                          TYPE   ,99$            ;
[illegible]                          CNVRT  ,89$            ;
[illegible]                          TYPE   ,91$            ;
[illegible]  174300                  TSTB   @TKCSR          ;WAIT FOR DONE.
[illegible]                          BPL    .-4
[illegible]  174274                  MOV    @TKCSR,-(SP)    ;
[illegible]                          BIC    #177[illegible],(SP) ;
[illegible]                          CMPB   #15,(SP)+       ;
[illegible]                          BNE    8$
[illegible]  174252                  MOV    (SP)+,@SWR
[illegible]  174254  2$:             TSTB   @TPCSR          ;
[illegible]                          BPL    .-4
[illegible]  177776  174246          MOV    -2(SP),@TPCSR   ;
[illegible]
[illegible]                                                 ;ERROR
[illegible]  177776  000060          [illegible] SP,#60     ;
[illegible]  177776  000067          [illegible] SP,#67     ;
[illegible]
[illegible]  174[illegible]          [illegible] @TKCSR,-(SP) ;
[illegible]                          [illegible]            ;
[illegible]                          [illegible]            ;
[illegible]  001302                  BNE    3$              ;
[illegible]  005104  5$:             TYPE   ,MCRLF          ;
[illegible]                          RTS    PC

[illegible]  051460  050127  99$:    .ASCIZ <377>/  SWR = /
[illegible]
                                 .EVEN
[illegible]  000001                  89$:   1
[illegible]  006     000             .BYTE  6,0
[illegible]                          90$:
[illegible]                          91$:   .WORD  0
[illegible]  036457  000057          [illegible]: .ASCIZ / = /
                                 .EVEN
[illegible]  000077                  MCRM:  .ASCIZ /[illegible]/
[illegible]  000                     MCRLF: .ASCIZ <15><12>
[illegible]  377     [illegible]     MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST/
[illegible]  377     [illegible]     MEPASS: .ASCIZ <377>/END PASS DZDVA-B/
[illegible]  377     [illegible]     MB:    .ASCIZ <377>/R/
[illegible]  377     [illegible]     MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
[illegible]  377     [illegible]     MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
[illegible]  377     [illegible]     MTSTPC: .ASCIZ <377>/TEST PC-/
[illegible]  377     [illegible]     MLOOP: .ASCIZ <377>/LOOP ON SELECTED TEST/
```

G03



```
                                                ;
                                                ;ROUTINE USED TO LOOP THROUGH UP TO EIGHT DZ11'S
                                                ;AND ALSO USED TO SET UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                                ;AND RUN THE SPECIFIED TESTS.  THIS ROUTINE IS ENTERED
                                                ;ONLY ONCE BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                                ;FIRST TIME.
                                                ;

001000          CYCLE:  MOV     #3,R0           ;ARE ANY DZ11'S TO BE TESTED
001002                  BNE     .NERR0          ;BR IF OK.
001004                  TSTB    @#177560        ;NO DZ11'S SELECTED!!
001010                  BPL     .               ;LOOP TILL THE SHOW.
001012                  MOVB    @#177560,R0     ;MODIFY CONT. SW.
001016                  BIC     #177774,R0      ;ISOLATE ONE DZ11 NUMBER
001022                  BEQ     .+4             ;IF ZERO ONE SELECTED.
001024                  SEC                     ;CLEAR PROG. CARRY BIT.
001026                  ROR     R0              ;CREATE POINTER
001030                  BCS     .               ;CATCH CARRY FROM RUN
001032                  ADD     .DZERR,R0       ;UPDATE ADDRESS POINTER.
001036                  MOV     (R0),R0
001040                  BNE     .+4             ;KEEP GOING, NOT A DZ11 TESTED FOR.
001042                  MOV     .DZERR,R0       ;RESET ADDRESS POINTER.
001046                  ASR     R0              ;LOOKING FOR ACTIVE DZ11
001050                  BCS     .               ;CLEAR PROG. CARRY.
001052                  ROL     R0              ;CREATE POINTER.
001054                  MOV     R0,-(SP)        ;SAVE WITH RECORD.
001056                  MOV     R0,.DZERR       ;GET ADDRESS POINTER.
001062                  MOV     (R0),R0         ;CREATE.
001064                  MOV     R0,.DZCSR
001070                  ADD     #10,R0
001074                  MOV     R0,.DZLPR
001100                  MOV     (SP)+,R0        ;ALL DONE?
001102                  BNE     .               ;BR IF NO.
001104                  MOV     #3,R0           ;RESTORE POINTER.
001110          30:     MOV     R0,-(SP)        ;CLEAR SILO CTRL. REG
001112                  CLR     @(R0)+          ;CLEAR XMIT CTRL.
001114                  CLR     @(R0)+          ;SET LINE PARAMETERS. 00-03
001116                  MOV     #BLOCK03,@(R0)+
001122                  MOV     #SYNC0A,@(R0)+
001126                  MOV     #LO4.00,@(R0)+  ;                     04-07
001132                  MOV     #SYNC0B,@(R0)+
001136                  MOV     #LO9.11,@(R0)+  ;                     09-11
001142                  MOV     #SYNC0C,@(R0)+
001146                  MOV     #L12.16,@(R0)+  ;                     12-15
001152                  MOV     #SYNC0D,@(R0)+
001156                  MOV     (SP)+,R0
001160                  MOVB    R0,@#DVSCR      ;SAVE CORE THIS WAY!
001164                  MOVB    @#DVSCR,DVSCRH  ;GET SYS CTRL. REG HIGH BYTE.
001172                  MOV     #DVSCRH,R0      ;GOT IT
001176                  MOVB    DVSCRH,DVRIC    ;GET NXT REC. CHAR REG.
001202                  INC     R0              ;GOT IT
001204                  MOV     R0,DVRIC        ;GET LN. PAR.REG.
001210                  MOVB    DVRIC,DVLCR     ;GOT IT
001214                  ADD     R0,R0           ;GET SEC. REG. SEL. REG.
001216                  MOV     R0,DVLCR        ;GOT IT
001222                  MOVB    DVLCR,DVSRS     ;GET HIGH BYTE.
001226                  ADD     R0,R0           ;GOT IT
001230                  MOV     R0,DVSRS        ;SEC. REG. ACCESS.
001234                  MOVB    DVSRS,DVSRSH    ;GOT IT
001240                  INC     DVSRSH
001244                  MOVB    DVSRSH,DVSRA
001250                  INC     DVSRA
```

```
RSX-11M V4.0 BL59   Task Builder   17-SEP-76  14:02  PAGE 35
[1,1]SYSUB.P11:     GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

001400 013737 001376 001400          MOV     @#VSRA,@#SFR    ;SPEC. FLT. REG.
001404 005737 001374                 TST     @#SFR           ;
001410 013737 001414 001402          MOV     @#VSFR,@#VNSR   ;NPR STAT  REG.
001416 013700 001414                 MOV     @#VSFR,R0       ;
001422 062700 000016 001404          ADD     #VNSR-VSFR,R0   ;RESERVED REG
001426 062737 000002 001404          ADD     #2,@#VNSR       ;

001432 013737 001354 001406          MOV     @#VVR,@#VVRLVL  ;PTY LVL
001436 013700 001354                 MOV     @#VVR,R0        ;
001442 062700 000004 001406          ADD     #4,R0           ;
001446 010037 001360                 MOV     R0,@#VVTVEC     ;TYP VEC
001452 062737 000002 001360          ADD     #2,@#VVTVEC     ;
001460 013700 001360                 MOV     @#VVTVEC,R0     ;TYP LVL
001464 062700 000037 001360          ADD     #37,R0

001470 012737 000003 001416          MOV     #3,@#LOAD       ;LOAD STAUS 00-03
001476 012737 000001 001420          MOV     #1,@#MASK       ;PREPARE MASK.
001504 012737 000002 001422          MOV     #2,@#CLK        ;PREPARE CLOCKS
001512 004737 006516                 JSR     PC,FIX          ;GO AND CALCULATE CONFIGURATION.

001516 012737 000003 001416          MOV     #3,@#LOAD       ;LOAD STAUS 00-03
001524 012737 000002 001420          MOV     #2,@#MASK       ;PREPARE MASK.
001532 012737 000001 001422          MOV     #1,@#CLK        ;PREPARE CLOCKS
001540 004737 006516                 JSR     PC,FIX          ;GO AND CALCULATE CONFIGURATION.

001544 012737 000017 001416          MOV     #15,@#LOAD      ;LOAD STAUS 00-03
001552 012737 000017 001420          MOV     #15,@#MASK      ;PREPARE MASK.
001560 012737 000002 001422          MOV     #2,@#CLK        ;PREPARE CLOCKS
001566 004737 006516                 JSR     PC,FIX          ;GO AND CALCULATE CONFIGURATION.
001572 012737 172516 001424          MOV     #SWR,@#ENA      ;
001600 000777                        BR      .               ;$

                             4$:
001602 005737 000042                 TST     @#42
001606 001002                        BNE     .               ;$
001610 005037 001214                 CLR     @#.TYPE         ;.MCR_F
001614 012737 001440 001216          MOV     #.INSTR,@#.MTSTN
001622 012737 001440 001210          MOV     #.INSTR,@#.PARAM
001630 005037 001212                 CLR     @#.TEMP
001634 012737 000200 001214          MOV     #200,@#.TSTNO
001642 012737 001226 001216          MOV     #TSTNO,@#.MTSTN
001650 000000                .BYTE   0
001651 000001                .BYTE   1
001652 012700 007256                 MOV     #TST1,R0
001656 022710                5$:     CMP     (PC)+,(R0)
001660 012737                        MOV     (PC)+,@(PC)+
001662 001016                        BNE     6$
001664 023760 001226 000002          CMP     TSTNO,2(R0)
001672 001011                        BNE     6$
001674 022760 001226 000004          CMP     #TSTNO,4(R0)
001702 001006                        BNE     6$
001704 010037 001214                 MOV     R0,@#.RETURN
```

```
                                TYPE    ,MCRLF
                                BR      5$
                        6$:     TST     (R0)+
                                CMP     R0,#TLAST+10
                                BNE     5$
                                TYPE    ,MOM
                                BR      4$
                        7$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
                        9$:     JMP     @RETURN         ;GO START TESTING.

                        FIX.CO: MOV     (R0),R3         ;GET PARAMETERS.
                                BIC     #^C<1400>,R3    ;CLEAR JUNK.
                                TST     R3              ;TEST FOR EIGHT BITS.
                                BNE     1$              ;BR IF NOT 8 BITS.
                                CLRB    (R1)            ;SET
                                MOVB    #8.,R2          ;
                                BR      4$
                        1$:     CMP     #400,R3         ;CHECK FOR SEVEN BITS.
                                BNE     2$              ;BR IF NOT 7 BITS.
                                MOVB    #200,(R1)       ;
                                MOVB    #7.,(R2)        ;
                                BR      4$
                        2$:     CMP     #1000,R3        ;CHECK FOR SIX BITS.
                                BNE     3$              ;BR IF NOT SIX BITS.
                                MOVB    #300,(R1)       ;
                                MOVB    #6.,(R2)        ;
                                BR      4$
                        3$:     MOVB    #340,(R1)       ;IF NONE OF THE ABOVE: MUST BE 5 BITS.
                                MOVB    #5.,(R2)        ;
                        4$:     BIT     #PARBIT,(R0)    ;PARITY ENABLED?
                                BEQ     5$              ;IF =0, THEN NO PARITY.
                                INCB    (R2)            ;PLUS ONE TO THE CLOCK!
                        5$:     RTS     PC              ;

                                ;*ROUTINE USED TO "AUTO SIZE" THE DV11
                                ;*CSR AND VECTOR.
                                ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
                                ;*      ADDRESS RANGE (175000:175400)
                                ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
                                ;*      FLOATING VECTOR RANGE (300:770)
                                ;*

                        AUTO.SIZE:
                                RESET                   ;INSURE A BUS INIT.
                        CSRMAP: MOV     #DV.MAP,R2      ;LOAD MAP POINTER.
                        1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
                                CMP     #DV.END,R2      ;ALL DONE?
                                BNE     1$              ;BR IF NO
                                CLRB    DVNUM           ;SET OCTAL NUMBER OF DV11'S TO 0
                                MOV     #DV.MAP,R2
                                MOV     #^175000,R1     ;SET FOR FIRST ADDRESS TO BE TESTED
                                MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
                        2$:     TST     (R1)            ;IF DV11 DVSCR S/B 0
                                BNE     3$              ;IF NO DEV ; TRAP TO 4, IF NO BIT 8 THEN NO DV11
                                CMP     #177777,12(R1)  ;IF DV11 THEN DVSCR S B ALL 1'S ON INIT!
                                BNE     3$              ;BR IF NOT DV11
```

```
                                                TST     16(R1)          ;IF DV11 THEN RES/16 S/B ALL 0'S
                                                BNE     3$              ;BR IF NOT DV11
                                        ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DV11 CSR ADDRESS.
                                                MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
                                                TST     (R2)+           ;POP OVER VECTOR STORE AREA
                                                BIS     #226,(R2)+      ;SET LINE CARD 1 STAT AND SYNC
                                                BIS     #62,(R2)+       ;
                                                BIS     #226,(R2)+      ;SET LINE CARD 2 STAT AND SYNC
                                                BIS     #62,(R2)+       ;
                                                BIS     #226,(R2)+      ;SET LINE CARD 3 STAT AND SYNC
                                                BIS     #62,(R2)+       ;
                                                BIS     #226,(R2)+      ;SET LINE CARD 4 STAT AND SYNC
                                                BIS     #62,(R2)+       ;
                                                INCB    DVNUM           ;UPDATE DEVICE COUNTER
                                                CMPB    #10,DVNUM       ;ARE MAX. NO. OF DEV FOUND?
                                                BEQ     10$             ;YES DON'T LOOK FOR ANY MORE.
                                        3$:     ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
                                                CMP     #175400,R1
                                                BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
                                        10$:    MOV     #177777,(R2)+   ;TERMINATER.
                                                CLRB    DVACTV
                                                TSTB    DVNUM           ;WERE ANY DV11'S FOUND AT ALL?
                                                BEQ     5$              ;ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
                                                MOVB    DVNUM,R1
                                                MOVB    R1,SAVNUM       ;SAVE NUMBER OF DEVICES
                                        4$:     CLC
                                                ROLB    DVACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
                                                INCB    DVACTV          ;SET THE BIT
                                                DEC     R1
                                                BNE     4$              ;BR IF MORE TO GENERATE
                                                MOV     #6,@#4          ;RESTORE TRAP VECTOR
                                                MOVB    DVACTV,SAVACT   ;SAVE ACTIVE REGISTER
                                                JMP     VECMAP          ;GO FIND THE VECTOR NOW.
                                        5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO DV11'S FOUND.
                                                CLR     R0              ;MAKE DATA LIGHTS ZERO
                                                HALT                    ;STOP THE SHOW
                                                BR      .-2             ;DISABLE CONT. SW.
                                        6$:     MOV     #3$,(SP)        ;ENTERED BY NON-EXISTANT TIME-OUT.
                                                RTI                     ;RETURN TO MAINSTREAM

                                        VECMAP: MOV     #340,@#22       ;SET IOT TRAP PRIO TO 7
                                                MOV     #4$,@#20        ;SET IOT TRAP VECTOR
                                                MOV     #DV.MAP,R2      ;SET SOFTWARE POINTER
                                                MOV     #300,R0         ;FLOATING VECTORS START HERE.
                                                MOV     #302,R1         ;PC OF IOT INSTR.
                                        1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
                                                MOV     #4,(R1)+        ;WITH .+2: IOT
                                                CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
                                                CMP     R1,#1000
                                                BLOS    1$              ;BR IF MORE TO FILL
                                                MOVB    DVACTV,TEMP1    ;STORE TEMPORALLY
                                        2$:     ROR     TEMP1           ;BRING OUT A BIT
                                                BCC     5$              ;BR IF ALL DONE
                                                CLR     PS              ;ZERO CPU PRIO
                                                MOV     #BIT9+BIT7+BIT6,@(R2)
                                                CLR     R0              ;ATTEMPT TO FORCE AN INTERUPT
```

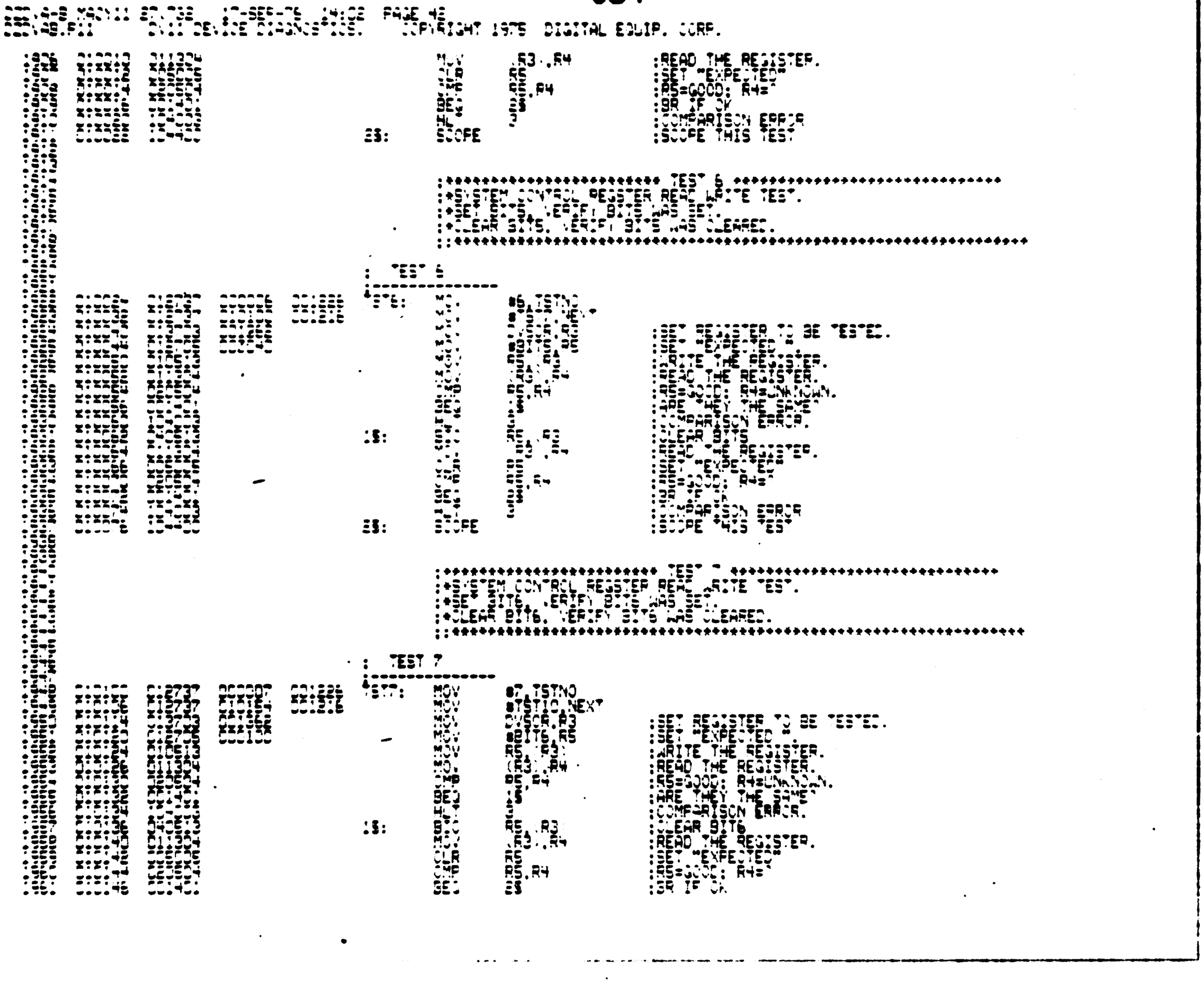
```
[illegible]  MACY11 27.732  12-SEP-76  14:02  PAGE 38
[illegible]  GENERAL UTILITIES .TYPE OUT,ERROR,SCOPE,ETC.)

[illegible]  005300                          INC     R0              ;STALL
[illegible]  001376                          BNE     .-2             ;       FOR TIME TO INTERUPT
[illegible]  052762  000300  000002          BIS     #300,2(R2)      ;NO INTERUPT ASSUME 300 AND FIX 5/11 LATER
[illegible]  042772  177700  000000  3$:     BIC     #1C(BIT9),3(R2)
[illegible]  005072  000000                  CLR     @(R2)
[illegible]  062702  000024                  ADD     #24,R2          ;POP SOFTWARE POINTER
[illegible]  000752                          BR      2$              ;KEEP GOING
[illegible]  051662  000002          4$:     BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
[illegible]  042763  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
[illegible]                                  CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
[illegible]  012716  007212                  MOV     #3$,(SP)        ;SET FOR RETURN
[illegible]                                  RTI
[illegible]                          5$:     RTS     PC              ;ALL DONE WITH "AUTO SIZING"
```

```
                                        ;********************** TEST 1 ****************************
                                        ;*VERIFY THAT ADDRESSING DEVICE DOES *NOT* CAUSE
                                        ;*A TIME-OUT TRAP.
                                        ;;*************************************************************

                                ;  TEST 1
                                ;--------------
007256  012737  000001  001226  TST1:   MOV     #1,TSTNO
007264  012737  007366  001216          MOV     #TST2,NEXT
007272  012737  007324  001220          MOV     #1$,LOCK
007300  012700  000010                  MOV     #8.,R0          ;SET FOR MAX. 8 PRI. REGISTERS
007304  013701  001362                  MOV     DVSCR,R1        ;GET FIRST PRI. ADDRESS
007310  012737  007360  000004          MOV     #2$,4           ;SET FOR TIME-OUT TRAP.
007316  012737  000340  000006          MOV     #340,6          ;SAFE GUARD.
007324  005711                  1$:     TST     (R1)            ;REFERENCE THE ADDRESS.
007326  000240                          NOP                     ;STALL
007330  000240                          NOP                     ;     FOR TIME.
007332  104401                          SCOP1                   ;IF SW09=1; GOTO 1$
007334  062701  000002                  ADD     #2,R1           ;UPDATE TO NEXT ADDRESS.
007340  005300                          DEC     R0              ;ARE ALL ADDRESS CHECKED?
007342  001370                          BNE     1$              ;BR IF NO.
007344  012737  000006  000004          MOV     #6,4            ;RESET TRAP ZONE.
007352  005037  000006                  CLR     @#6             ;
007356  104400                          SCOPE                   ;SCOPE THIS TEST
007360  011602                  2$:     MOV     (SP),R2         ;SAVE THE TRAP PC
007362  104001                          HLT     1               ;REPORT TIME-OUT TRAP
007364  000002                          RTI                     ;RETURN TO MAIN PROGRAM


                                        ;********************** TEST 2 ****************************
                                        ;*PRIMARY REGISTER ADDRESSING TEST
                                        ;*LOAD EACH PRIMARY REGISTER WITH A
                                        ;*DIFFERENT NUMBER AND VERIFY EACH
                                        ;*WAS INDIVIDUALLY ADDRESSED.
                                        ;;*************************************************************

                                ;  TEST 2
                                ;--------------
007366  012737  000002  001226  TST2:   MOV     #2,TSTNO
007374  012737  007620  001216          MOV     #TST3,NEXT
007402  012737  007436  001220          MOV     #1$,LOCK
007410  012700  007600                  MOV     #3$,R0          ;SET DATA TABLE POINTER
007414  013703  001362                  MOV     DVSCR,R3        ;SET DV POINTER
007420  005013                          CLR     (R3)            ;START REG AT ZERO
007422  005077  171744                  CLR     @DVSRS          ;ZERO SEC. REG SEL.
007426  005077  171744                  CLR     @DVSRA          ;ZERO SEC REG ACCESS
007432  012702  000010                  MOV     #8.,R2          ;SET FOR EIGHT PRIMARY REGISTERS.
007436  011005                  1$:     MOV     (R0),R5         ;PUT DATA INTO EXPECTED
007440  010513                          MOV     R5,(R3)         ;WRITE EXPECTED INTO DV REGISTER
007442  011304                          MOV     (R3),R4         ;READ REGISTER INTO FOUND LOC
007444  020504                          CMP     R5,R4           ;DOES EXPECTED=RECEIVED?
007446  001401                          BEQ     64$             ;BR IF YES
007450  104003                          HLT     3               ;THIS IS A DATA ERROR *NOT* A DUEL ADDRESSING ERROR. NO?
007452  104401                  64$:    SCOP1                   ;SW09=1?
007454  022023                          CMP     (R0)+,(R3)+     ;POP DATA POINTERS AND HRDW POINTER
007456  020337  001402                  CMP     R3,DVNSR        ;DON'T DO THE DVNSR!
```

```
1794  007462  001002                          BNE     4$              ;BR IF NOT DVNSR
1795  007464  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS AROUND DVNSR
1796  007466  005302                          DEC     R2              ;UPDATE REGISTER COUNTER
1797  007470  020337  001370          4$:     CMP     R3,DVLCR        ;DON'T DO THE DVLCR!
1798  007474  001002                          BNE     6$              ;BR IF NOT THE DVLCR
1799  007476  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS AROUND THE DVLCR
1800  007500  005302                          DEC     R2              ;UPDATE THE REGISTER COUNTER
1801  007502  005302                  6$:     DEC     R2              ;DITTO
1802  007504  001354                          BNE     1$              ;BR IF MORE TO GO
1803                                  ;CHECK DUEL ADDRESSING.....
1804  007506  012700  007600                  MOV     #3$,R0          ;SET DATA POINTER
1805  007512  013703  001362                  MOV     DVSCR,R3        ;SET HRDW POINTER
1806  007516  012737  007530  001220          MOV     #2$,LOCK        ;SET IF SW09=1
1807  007524  012702  000010                  MOV     #8.,R2          ;SET EIGHT PRIMARY REGISTERS
1808  007530  011005                  2$:     MOV     (R0),R5         ;LOAD DATA INTO EXPECTED
1809  007532  011304                          MOV     (R3),R4         ;READ THE DV REGISTER
1810  007534  020504                          CMP     R5,R4           ;DOES THE DATA COMPARE
1811  007536  001401                          BEQ     55$             ;BR IF OK
1812  007540  104003                          HLT     3               ;NOW THIS WAS A DUEL ADDRESSING ERROR.
1813  007542  104401                  55$:    SCOP1                   ;SW09=1?
1814  007544  022023                          CMP     (R0)+,(R3)+     ;POP POINTERS
1815  007546  020337  001402                  CMP     R3,DVNSR        ;DON'T DO THE DVNSR
1816  007552  001002                          BNE     5$              ;BR IF NOT DVNSR
1817  007554  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS
1818  007556  005302                          DEC     R2              ;SET REG COUNTER
1819  007560  020337  001370          5$:     CMP     R3,DVLCR        ;DON'T DO THE DVLCR
1820  007564  001002                          BNE     7$              ;BR IF NOT DVLCR
1821  007566  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS
1822  007570  005302                          DEC     R2              ;SET REG POINTER
1823  007572  005302                  7$:     DEC     R2              ;DITTO
1824  007574  001355                          BNE     2$              ;BR IF MORE TO GO
1825  007576  104400                          SCOPE                   ;SCOPE THIS TEST
1826  007600  000010                  3$:     .WORD   000010          ;DVSCR
1827  007602  000000                          .WORD   000000          ;DVRIC
1828  007604  000000                          .WORD   SKIP            ;DVLCR
1829  007606  001400                          .WORD   001400          ;DVSRS
1830  007610  000300                          .WORD   000300          ;DVSRA
1831  007612  100000                          .WORD   100000          ;DVSFR
1832  007614  000000                          .WORD   SKIP            ;DVNSR
1833  007616  000060                          .WORD   000060          ;RESV16
1834
1835
1836                                          ;************************ TEST 3 ******************************
1837                                          ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
1838                                          ;*SET BIT2, VERIFY BIT2 WAS SET.
1839                                          ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
1840                                          ;:**********************************************************
1841
1842                                  ;  TEST 3
1843                                  ;--------------
1844  007620  012737  000003  001226  TST3:   MOV     #3,TSTNO
1845  007626  012737  007674  001216          MOV     #TST4,NEXT
1846  007634  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
1847  007640  012705  000004                  MOV     #BIT2,R5        ;SET "EXPECTED ".
1848  007644  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
1849  007646  011304                          MOV     (R3),R4         ;READ THE REGISTER.
```

```
                                                ;R5=GOOD; R4=UNKNOWN.
                                                ;ARE THEY THE SAME
                                                ;COMPARISON ERROR.
                                                ;CLEAR BIT2
                                                ;READ THE REGISTER.
                                                ;R4= EXPECTED
                                                ;R4=GOOD; R4=?
                                                ;CLEAR BIT
                                                ;COMPARISON ERROR
                                                ;SKIP THIS TEST

                ;********************** TEST 4 ****************************
                ;SYSTEM CONTROL REGISTER READ/WRITE TEST.
                ;CLEAR BIT3, VERIFY BIT3 WAS SET.
                ;SET BIT3, VERIFY BIT3 WAS CLEARED.
                ;**********************************************************

        ; TEST 4
        ;--------
T4:                                             ;SET REGISTER TO BE TESTED.
                                                ;SET "EXPECTED"
                                                ;WRITE THE REGISTER.
                                                ;READ THE REGISTER.
                                                ;R5=GOOD; R4=UNKNOWN.
                                                ;ARE THEY THE SAME
                                                ;COMPARISON ERROR.
                                                ;CLEAR BIT3
                                                ;READ THE REGISTER.
                                                ;R4= EXPECTED
                                                ;R4=GOOD; R4=?
                                                ;CLEAR BIT
                                                ;COMPARISON ERROR
                                                ;SKIP THIS TEST

                ;********************** TEST 5 ****************************
                ;SYSTEM CONTROL REGISTER READ/WRITE TEST.
                ;SET BIT4, VERIFY BIT4 WAS SET.
                ;CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
                ;**********************************************************

        ; TEST 5
        ;--------
T5:     MOV     #5,TSTNO
        MOV     #TST6,NEXT
        MOV     DVSCSR,R3               ;SET REGISTER TO BE TESTED.
        MOV     #BIT4,R5                ;SET "EXPECTED"
        MOV     R5,(R3)                 ;WRITE THE REGISTER.
        MOV     (R3),R4                 ;READ THE REGISTER.
        CMP     R5,R4                   ;R5=GOOD; R4=UNKNOWN.
        BEQ     10$                     ;ARE THEY THE SAME
        JMP     3$                      ;COMPARISON ERROR.
3$:     BIC     R5,(R3)                 ;CLEAR BIT4
```

C04



```
                                        HLT     2               ;COMPARISON ERROR
                                88:     SCOPE                   ;SCOPE THIS TEST

                                        ;************************ TEST 10 ******************************
                                        ;*TEST THAT BIT 7,RECEIVER INTERUPT)
                                        ;*CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
                                        ;*IS NOT SET.
                                        ;*THEN VERIFY THAT BIT 7 CAN BE WRITTEN WHEN
                                        ;*BIT 9 IS SET.
                                        ;*************************************************************

                                ;  TEST 10
                                ;--------------
                                T10:    MOV     #10,TSTNO
                                        MOV     #4,R3           ;SET REGISTER POINTER INTO R3
                                        CLR     R4              ;CLEAR EXPECTED RESULT
                                        BIS     R3,R4           ;ATTEMPT TO SET BIT 7
                                        CMP     R3,R4           ;READ BACK REGISTER
                                        BEQ     .               ;BIT 7 SHOULD NOT BE SET
                                        MOV     R3,@#SCR        ;MOVE TO SCR
                                        BIS     #BIT7+BIT9,R5   ;SET BIT 7+BIT9 IN THE EXPECTED
                                        MOV     R3,R4           ;
                                        BIS     R5,R4           ;READ REGISTER
                                        CMP     R3,R4           ;BIT 9 AND BIT 7 SET
                                        BEQ     .               ;IF OK
                                        MOV     R3,@#SCR        ;MOVE TO SCR
                                        CLR     R4              ;CLEAR EXPECTED
                                        BIS     R5,R4           ;SET BIT 7 IN EXPECTED
                                        CMP     R3,R4           ;READ REGISTER
                                        BEQ     .               ;BIT 7 SHOULD BE SET
                                        MOV     R3,@#SCR        ;MOVE TO SCR
                                        CLR     R4              ;CLEAR EXPECTED
                                        BIS     R5,R4           ;SET BIT 7 IN EXPECTED
                                        BIC     #BIT9,R4        ;CLEAR BIT 9 IN EXPECTED
                                        CMP     R3,R4           ;READ REGISTER
                                        BEQ     .               ;BIT 9 SHOULD BE CLEARED
                                        MOV     R3,@#SCR        ;MOVE TO SCR
                                88:     SCOPE                   ;SCOPE TEST

                                        ;************************ TEST 11 ******************************
                                        ;*TEST THAT BIT 7 (RECEIVER INTERUPT)
                                        ;*CAN BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
                                        ;*IS SET.
                                        ;*************************************************************

                                ;  TEST 11
                                ;--------------
                                T11:    MOV     #11,TSTNO
                                        MOV     #4,R3           ;SET REGISTER TO BE TESTED.
                                        CLR     R4              ;CLEAR EXPECTED
                                        BIS     R5,R4           ;SET EXPECTED
                                        CMP     R3,R4           ;WRITE THE REGISTER.
                                        BEQ     .               ;READ THE REGISTER.
                                        MOV     R3,@#SCR        ;MOVE TO SCR.
                                        BEQ     T12             ;GOOD. RETURN.
                                        MOV     R3,@#SCR        ;MOVE TO SCR.
                                        CMP     R3,R4           ;READ ERROR AGAIN.
                                        BEQ     .               ;ERROR BIT 9
                                        MOV     R3,@#SCR        ;MOVE TO REGISTER.
```

```
                                                        ;SET "EXPECTED"
                                                        ;R5=GOOD; R4=
                                                        ;BR IF OK
                                                        ;COMPARISON ERROR
                                28:     SCOPE           ;SCOPE THIS TEST

                                        ;************************ TEST 12 ******************************
                                        ;*SYSTEM CONTROL REGISTER READ WRITE TEST.
                                        ;*SET BIT9, VERIFY BIT9 WAS SET.
                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
                                        ;***********************************************************

                                ;  TEST 12
                                ;--------------
                                TST12:  MOV     #12,TSTNO
                                        MOV     #TST13,NEXT
                                        MOV     #SCSR,R3        ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT9,R5        ;SET "EXPECTED".
                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                        BEQ     .+4             ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                16:     BIC     R5,(R3)         ;CLEAR BIT9
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=
                                        BEQ     .+4             ;BR IF OK
                                        HLT     3               ;COMPARISON ERROR
                                18:     SCOPE                   ;SCOPE THIS TEST

                                        ;************************ TEST 13 ******************************
                                        ;*SYSTEM CONTROL REGISTER READ WRITE TEST.
                                        ;*SET BIT10, VERIFY BIT10 WAS SET.
                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
                                        ;***********************************************************

                                ;  TEST 13
                                ;--------------
                                TST13:  MOV     #13,TSTNO
                                        MOV     #TST14,NEXT
                                        MOV     #SCSR,R3        ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT10,R5       ;SET "EXPECTED".
                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                        BEQ     .+4             ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                16:     BIC     R5,(R3)         ;CLEAR BIT10
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=
                                        BEQ     .+4             ;BR IF OK
                                        HLT     3               ;COMPARISON ERROR
```

```
 013400  104400                  28:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 14 ****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT12. VERIFY BIT12 WAS SET.
                                        ;*CLEAR BIT12. VERIFY BIT12 WAS CLEARED.
                                        ;************************************************************

                                ;  TEST 14
                                ;--------------
 013402                         ST14:   MOV     #ST14,TSTNO
 013410                                 MOV     #ST15,NEXT
 013416  012705  177570                 MOV     #SYSCR,R5       ;SET REGISTER TO BE TESTED.
 013422  012704  010000                 MOV     #BIT12,R4       ;SET "EXPECTED"
 013426  010514                         MOV     R5,(R4)         ;WRITE THE REGISTER.
 013430  011403                         MOV     (R4),R3         ;READ THE REGISTER.
 013432  020304                         CMP     R3,R4           ;R3=GOOD; R4=UNKNOWN.
 013434                                 ;ARE THEY THE SAME?
 013434                                 ;COMPARISON ERROR.
 013434  042704  010000         10:     BIC     #BIT12,R4       ;CLEAR BIT12
 013440  011403                         MOV     (R4),R3         ;READ THE REGISTER.
 013442                                 ;SHOULD "EXPECTED"
 013442                                 ;R3=GOOD; R4=?
 013442  020304                         CMP     R3,R4           ;BR IF OK
 013444                                 ;COMPARISON ERROR
 013444  104400                 11:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 15 ****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT13. VERIFY BIT13 WAS SET.
                                        ;*CLEAR BIT13. VERIFY BIT13 WAS CLEARED.
                                        ;************************************************************

                                ;  TEST 15
                                ;--------------
 013446                         ST15:   MOV     #ST15,TSTNO
 013454                                 MOV     #ST16,NEXT
 013462  012705  177570                 MOV     #SYSCR,R5       ;SET REGISTER TO BE TESTED.
 013466  012704  020000                 MOV     #BIT13,R4       ;SET "EXPECTED"
 013472  010514                         MOV     R5,(R4)         ;WRITE THE REGISTER.
 013474  011403                         MOV     (R4),R3         ;READ THE REGISTER.
 013476  020304                         CMP     R3,R4           ;R3=GOOD; R4=UNKNOWN.
 013500                                 ;ARE THEY THE SAME?
 013500                                 ;COMPARISON ERROR.
 013500  042704  020000         10:     BIC     #BIT13,R4       ;CLEAR BIT13
 013504  011403                         MOV     (R4),R3         ;READ THE REGISTER.
 013506                                 ;SHOULD "EXPECTED"
 013506                                 ;R3=GOOD; R4=?
 013506  020304                         CMP     R3,R4           ;BR IF OK
 013510                                 ;COMPARISON ERROR
 013510  104400                 11:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 16 ****************************
```

```
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*BIT3+BIT0. VERIFY BIT3+BIT0 WAS SET.
                                        ;*CLEAR BIT3+BIT0. VERIFY BIT3+BIT0 WAS CLEARED.
                                        ;*************************************************************

                                ;  TEST 16
                                ;--------------
                                ST16:   MOV     #16,TSTNO
                                        MOV     #177570,R3              ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT3+BIT0,R5           ;SET "EXPECTED"
                                        MOV     R5,(R3)                 ;WRITE THE REGISTER.
                                        MOV     (R3),R4                 ;READ THE REGISTER.
                                        CMP     R5,R4                   ;REG=GOOD; R4=UNKNOWN.
                                        BEQ     .+4                     ;ARE THEY THE SAME?
                                        IOT                             ;COMPARISON ERROR.
                                        CLR     (R3)                    ;CLEAR BIT3+BIT0
                                        MOV     (R3),R4                 ;READ THE REGISTER.
                                        TST     R4                      ;SET "EXPECTED"
                                        BEQ     .+4                     ;R5=GOOD; R4=?
                                        IOT                             ;BR IF OK
                                        JMP     ST17                    ;COMPARISON ERROR
                                ;       SCOPE                           ;SCOPE THIS TEST

                                        ;************************ TEST 17 ********************************
                                        ;*TEST THAT BIT 15/NPR STATUS OVERFLOW INTERRUPT
                                        ;*CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTANCE
                                        ;*BIT) IS NOT SET.
                                        ;*THEN VERIFY THAT BIT 15 CAN BE WRITTEN WHEN
                                        ;*BIT 9 IS SET.
                                        ;*************************************************************

                                ;  TEST 17
                                ;--------------
                                ST17:   MOV     #17,TSTNO
                                        MOV     #177570,R3              ;SET REGISTER POINTER INTO R3
                                        MOV     (R3),R4                 ;READ REGISTER
                                        MOV     #BIT15,(R3)             ;WRITE TEMP REGISTER WITH BIT 15
                                        MOV     (R3),R4                 ;READ BACK REGISTER
                                        BIT     #BIT15,R4               ;BIT 15 SHOULD NOT BE SET
                                        BEQ     .+4                     ;OVSCR NOT SET
                                        IOT
                                        MOV     #BIT15+BIT9,R5          ;SET BIT 15+BIT9 IN THE DATA
                                        MOV     R5,(R3)
                                        MOV     (R3),R4                 ;READ REGISTER
                                        CMP     R5,R4                   ;IS BIT 9 AND BIT 15 SET?
                                        BEQ     .+4                     ;BR IF OK
                                        IOT                             ;OVSCR ERROR
                                        BIC     #BIT15,R5               ;CLEAR BIT 15
                                        BIS     #BIT15,R3               ;DITTO
                                        MOV     (R3),R4                 ;READ REGISTER
                                        CMP     R5,R4                   ;COMPARE OK?
                                        BEQ     .+4                     ;BR IF YES
                                        IOT                             ;OVSCR ERROR
                                ;       SCOPE                           ;SCOPE THIS TEST
```



```
                        ;*********************** TEST 20 ***************************
                        ;*TEST THAT BIT8 IN DVSCR CLEARS.
                        ;*BIT* OF DVSCR.
                        ;;***********************************************************

                        ;  TEST 20
                        ;--------------
                        TST20:  MOV     #20,TSTNO
                                MOV     #TST21,NEXT
                                MOV     DVSCR,R3        ;GET POINTER
                                CLR     R5              ;CLEAR
                                MOV     #-1,R3          ;WRITE
                                MOV     R3,R4           ;READ
                                CMP     R5,R4           ;WRITE TIME
                                BEQ     .+6             ;READ OKAY
                                MOV     #3,R4           ;BIT8 ONLY SET
                                BR      .+4             ;OR IF YES
                                MOV     #4,R4           ;BIT8 NOT ONLY BIT SET OR B ISN'T SET!
                                CLR     (R3)            ;CLEAR REGC
                        1$:     SCOPE                   ;SCOPE

                        ;*********************** TEST 21 ***************************
                        ;*RECEIVER INTERUPT CHARACTER REGISTER TEST
                        ;*VERIFY THAT RECEIVER INTERUPT CHARACTER REGISTER CANNOT BE WRITTEN.
                        ;*WRITE RECEIVER INTERUPT CHARACTER REGISTER WITH ALL 1'S
                        ;*AND VERIFY THAT ALL 0'S ARE READ BACK.
                        ;;***********************************************************

                        ;  TEST 21
                        ;--------------
                        TST21:  MOV     #21,TSTNO
                                MOV     #TST22,NEXT
                                MOV     DVRIC,R3
                                CLR     R5              ;SET CLR REGISTER.
                                MOV     #-1,R3          ;SET EXPECTED ZERO.
                                MOV     R3,R4           ;WRITE REGISTER WITH ALL 1'S
                                CMP     R5,R4           ;READ THE REGISTER.
                                BEQ     .+6             ;IS THE REGISTER EQUAL TO ZERO.
                                MOV     #3,R4           ;BR IF OK.
                        1$:     SCOPE                   ;REGISTER NOT ZERO.
                                                        ;SCOPE THIS TEST.

                        ;*********************** TEST 22 ***************************
                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                        ;*SET BIT9, VERIFY BIT9 WAS SET.
                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
                        ;;***********************************************************

                        ;  TEST 22
                        ;--------------
                        TST22:  MOV     #22,TSTNO
                                MOV     #TST23,NEXT
                                MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
                                                        ;SET "EXPECTED "
                                                        ;WRITE THE REGISTER.
                                                        ;READ THE REGISTER.
                                                        ;CLEAR UNWANTED BITS
                                                        ;R5=GOOD; R4=UNKNOWN.
                                                        ;ARE THEY THE SAME?
                                                        ;COMPARISON ERROR.
                                                        ;CLEAR BIT9
                                                        ;READ THE REGISTER.
                                                        ;CLEAR UNWANTED BITS
                                                        ;SET "EXPECTED"
                                                        ;R5=GOOD; R4=?
                                                        ;BR IF OK
                                                        ;COMPARISON ERROR
                                                        ;SCOPE THIS TEST

                        ;********************** TEST 23 ****************************
                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                        ;*SET BIT10, VERIFY BIT10 WAS SET.
                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
                        ;;*********************************************************

                        ;  TEST 23
                        ;--------------
000000  012737  000023  001226  TST23:  MOV     #23,TSTNO
000006  012737  000000  001216          MOV     #0,ERRCNT
000014  012703  176500                  MOV     #DVLCR,R3       ;SET REGISTER TO BE TESTED.
000020  012713  002000                  MOV     #BIT10,(R3)     ;SET "EXPECTED "
000024  011323                          MOV     (R3),(R3)       ;WRITE THE REGISTER.
000026  011304                          MOV     (R3),R4         ;READ THE REGISTER.
000030  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
000034  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
000036  001403                          BEQ     1$              ;ARE THEY THE SAME?
000040  104003                          HALT    3               ;COMPARISON ERROR.
000042  010313                  1$:     MOV     R3,(R3)         ;CLEAR BIT10
000044  011304                          MOV     (R3),R4         ;READ THE REGISTER.
000046  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
000052  020504                          CMP     R5,R4           ;SET "EXPECTED"
000054  001403                          BEQ     2$              ;R5=GOOD; R4=?
000056  104003                          HALT    3               ;BR IF OK
000060  000240                          NOP                     ;COMPARISON ERROR
000062  104400                  2$:     SCOPE                   ;SCOPE THIS TEST

                        ;********************** TEST 24 ****************************
                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                        ;*SET BIT11, VERIFY BIT11 WAS SET.
                        ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
                        ;;*********************************************************

                        ;  TEST 24
                        ;--------------
000064  012737  000024  001226  TST24:  MOV     #24,TSTNO
000072  012737  001306  001216          MOV     #TST25,NEXT
000100  012703  176500                  MOV     #DVLCR,R3       ;SET REGISTER TO BE TESTED.
```

```
                                MOV     #BIT11,R5       ;SET "EXPECTED ".
                                MOV     R5,(R3)         ;WRITE THE REGISTER.
                                MOV     (R3),R4         ;READ THE REGISTER.
                                BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
                                CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                BEQ     1$              ;ARE THEY THE SAME?
                                HLT     3               ;COMPARISON ERROR.
                        1$:     BIC     R5,(R3)         ;CLEAR BIT11
                                MOV     (R3),R4         ;READ THE REGISTER.
                                BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
                                CLR     R5              ;SET "EXPECTED"
                                CMP     R5,R4           ;R5=GOOD; R4=?
                                BEQ     2$              ;BR IF OK
                                HLT     3               ;COMPARISON ERROR
                        2$:     SCOPE                   ;SCOPE THIS TEST


                                ;************************ TEST 25 ******************************
                                ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                ;*SET BIT12, VERIFY BIT12 WAS SET.
                                ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.
                                ;;*************************************************************

                                ;  TEST 25
                                ;--------------
011326  012737  000025  001226  TST25:  MOV     #25,TSTNO
011334  012737  011412  001216          MOV     #TST26,NEXT
011342  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
011346  012705  010000                  MOV     #BIT12,R5       ;SET "EXPECTED ".
011352  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011354  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011356  042704  000053                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011362  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011364  001401                          BEQ     1$              ;ARE THEY THE SAME?
011366  104003                          HLT     3               ;COMPARISON ERROR.
011370  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT12
011372  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011374  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011400  005005                          CLR     R5              ;SET "EXPECTED"
011402  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011404  001401                          BEQ     2$              ;BR IF OK
011406  104003                          HLT     3               ;COMPARISON ERROR
011410  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                ;************************ TEST 26 ******************************
                                ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                ;*SET BIT13, VERIFY BIT13 WAS SET.
                                ;*CLEAR BIT13, VERIFY BIT13 WAS CLEARED.
                                ;;*************************************************************

                                ;  TEST 26
                                ;--------------
011412  012737  000026  001226  TST26:  MOV     #26,TSTNO
011420  012737  011476  001216          MOV     #TST27,NEXT
011426  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
011432  012705  020000          MOV     #BIT13,R5       ;SET "EXPECTED ".
011436  010513                  MOV     R5,(R3)         ;WRITE THE REGISTER.
011440  011304                  MOV     (R3),R4         ;READ THE REGISTER.
011442  042704  000063          BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011446  020504                  CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011450  001401                  BEQ     1$              ;ARE THEY THE SAME?
011452  104003                  HLT     3               ;COMPARISON ERROR.
011454  040513          1$:     BIC     R5,(R3)         ;CLEAR BIT13
011456  011304                  MOV     (R3),R4         ;READ THE REGISTER.
011460  042704  000063          BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011464  005005                  CLR     R5              ;SET "EXPECTED"
011466  020504                  CMP     R5,R4           ;R5=GOOD; R4=?
011470  001400                  BEQ     2$              ;BR IF OK
011472  104003                  HLT     3               ;COMPARISON ERROR
011474  104400          2$:     SCOPE                   ;SCOPE THIS TEST


                                ;********************** TEST 27 ****************************
                                ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                ;*SET BIT14, VERIFY BIT14 WAS SET.
                                ;*CLEAR BIT14, VERIFY BIT14 WAS CLEARED.
                                ;;**********************************************************

                                ;  TEST 27
                                ;--------------
011476  012737  000027  001226  TST27:  MOV     #27,TSTNO
011504  012737  011562  001216          MOV     #TST30,NEXT
011512  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
011516  012705  040000                  MOV     #BIT14,R5       ;SET "EXPECTED ".
011522  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011524  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011526  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011532  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011534  001401                          BEQ     1$              ;ARE THEY THE SAME?
011536  104003                          HLT     3               ;COMPARISON ERROR.
011540  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT14
011542  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011544  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011550  005005                          CLR     R5              ;SET "EXPECTED"
011552  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011554  001401                          BEQ     2$              ;BR IF OK
011556  104003                          HLT     3               ;COMPARISON ERROR
011560  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                ;********************** TEST 30 ****************************
                                ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                ;*SET BIT0, VERIFY BIT0 WAS SET.
                                ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
                                ;;**********************************************************

                                ;  TEST 30
                                ;--------------
011562  012737  000030  001226  TST30:  MOV     #30,TSTNO
011570  012737  011636  001216          MOV     #TST31,NEXT
011576  013703  001372                  MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
```

```
011600  012705  000001                MOV     #BIT0,R5        ;SET "EXPECTED ".
011604  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
011606  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011610  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011612  001401                        BEQ     1$              ;ARE THEY THE SAME?
011614  104003                        HLT     3               ;COMPARISON ERROR.
011616  040513                1$:     BIC     R5,(R3)         ;CLEAR BIT0
011620  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011622  005005                        CLR     R5              ;SET "EXPECTED"
011624  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
011626  001401                        BEQ     2$              ;BR IF OK
011630  104003                        HLT     3               ;COMPARISON ERROR
011632  104400                2$:     SCOPE                   ;SCOPE THIS TEST


                                    ;********************** TEST 31 ****************************
                                    ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                    ;*SET BIT1, VERIFY BIT1 WAS SET.
                                    ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
                                    ;;**********************************************************

                                ;  TEST 31
                                ;--------------
011636  012737  000031  001226  TST31:  MOV     #31,TSTNO
011644  012737  011712  001216          MOV     #TST32,NEXT
011652  013703  001372                  MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
011656  012705  000002                  MOV     #BIT1,R5        ;SET "EXPECTED ".
011662  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
011664  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011666  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011670  001401                        BEQ     1$              ;ARE THEY THE SAME?
011672  104003                        HLT     3               ;COMPARISON ERROR.
011674  040513                1$:     BIC     R5,(R3)         ;CLEAR BIT1
011676  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011700  005005                        CLR     R5              ;SET "EXPECTED"
011702  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
011704  001401                        BEQ     2$              ;BR IF OK
011706  104003                        HLT     3               ;COMPARISON ERROR
011710  104400                2$:     SCOPE                   ;SCOPE THIS TEST


                                    ;********************** TEST 32 ****************************
                                    ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                    ;*SET BIT2, VERIFY BIT2 WAS SET.
                                    ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
                                    ;;**********************************************************

                                ;  TEST 32
                                ;--------------
011712  012737  000032  001226  TST32:  MOV     #32,TSTNO
011720  012737  011766  001216          MOV     #TST33,NEXT
011726  013703  001372                  MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
011732  012705  000004                  MOV     #BIT2,R5        ;SET "EXPECTED ".
011736  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
011740  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011742  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
```

```
 2466  011744  001401                        BEQ     1$              ;ARE THEY THE SAME?
 2467  011746  104003                        HLT     3               ;COMPARISON ERROR.
 2468  011750  040513                1$:     BIC     R5,(R3)         ;CLEAR BIT2
 2469  011752  011304                        MOV     (R3),R4         ;READ THE REGISTER.
 2470  011754  005005                        CLR     R5              ;SET "EXPECTED"
 2471  011756  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
 2472  011760  001401                        BEQ     2$              ;BR IF OK
 2473  011762  104003                        HLT     3               ;COMPARISON ERROR
 2474  011764  104400                2$:     SCOPE                   ;SCOPE THIS TEST
 2475
 2476
 2477                                        ;************************ TEST 33 ****************************
 2478                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
 2479                                        ;*SET BIT3, VERIFY BIT3 WAS SET.
 2480                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
 2481                                        ;:**********************************************************
 2482
 2483                                ;  TEST 33
 2484                                ;--------------
 2485  011766  012737  000033  001226 TST33:  MOV     #33,TSTNO
 2486  011774  012737  012042  001216         MOV     #TST34,NEXT
 2487  012002  013703  001372                 MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
 2488  012006  012705  000010                 MOV     #BIT3,R5        ;SET "EXPECTED ".
 2489  012012  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
 2490  012014  011304                         MOV     (R3),R4         ;READ THE REGISTER.
 2491  012016  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2492  012020  001401                         BEQ     1$              ;ARE THEY THE SAME?
 2493  012022  104003                         HLT     3               ;COMPARISON ERROR.
 2494  012024  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT3
 2495  012026  011304                         MOV     (R3),R4         ;READ THE REGISTER.
 2496  012030  005005                         CLR     R5              ;SET "EXPECTED"
 2497  012032  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
 2498  012034  001401                         BEQ     2$              ;BR IF OK
 2499  012036  104003                         HLT     3               ;COMPARISON ERROR
 2500  012040  104400                2$:      SCOPE                   ;SCOPE THIS TEST
 2501
 2502
 2503                                        ;************************ TEST 34 ****************************
 2504                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
 2505                                        ;*SET BIT8, VERIFY BIT8 WAS SET.
 2506                                        ;*CLEAR BIT9, VERIFY BIT8 WAS CLEARED.
 2507                                        ;:**********************************************************
 2508
 2509                                ;  TEST 34
 2510                                ;--------------
 2511  012042  012737  000034  001226 TST34:  MOV     #34,TSTNO
 2512  012050  012737  012116  001216         MOV     #TST35,NEXT
 2513  012056  013703  001372                 MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
 2514  012062  012705  000400                 MOV     #BIT8,R5        ;SET "EXPECTED ".
 2515  012066  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
 2516  012070  011304                         MOV     (R3),R4         ;READ THE REGISTER.
 2517  012072  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2518  012074  001401                         BEQ     1$              ;ARE THEY THE SAME?
 2519  012076  104003                         HLT     3               ;COMPARISON ERROR.
 2520  012100  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT8
 2521  012102  011304                         MOV     (R3),R4         ;READ THE REGISTER.
```

```
002522  012104  005005                        CLR     R5              ;SET "EXPECTED"
002523  012106  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
002524  012110  001401                        BEQ     2$              ;BR IF OK
002525  012112  104003                        HLT     3               ;COMPARISON ERROR
002526  012114  104400              2$:       SCOPE                   ;SCOPE THIS TEST
002527
002528
002529                                        ;************************ TEST 35 ****************************
002530                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
002531                                        ;*SET BIT9, VERIFY BIT9 WAS SET.
002532                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
002533                                        ;;**********************************************************************
002534
002535                                ;  TEST 35
002536                                ;--------------
002537  012116  012737  000035  001226 TST35:  MOV     #35,TSTNO
002540  012124  012737  012172  001216        MOV     #TST36,NEXT
002541  012132  013703  001372                MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
002542  012136  012705  001000                MOV     #BIT9,R5        ;SET "EXPECTED ".
002543  012142  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
002544  012144  011304                        MOV     (R3),R4         ;READ THE REGISTER.
002545  012146  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
002546  012150  001401                        BEQ     1$              ;ARE THEY THE SAME?
002547  012152  104003                        HLT     3               ;COMPARISON ERROR.
002550  012154  040513              1$:       BIC     R5,(R3)         ;CLEAR BIT9
002551  012156  011304                        MOV     (R3),R4         ;READ THE REGISTER.
002552  012160  005005                        CLR     R5              ;SET "EXPECTED"
002553  012162  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
002554  012164  001401                        BEQ     2$              ;BR IF OK
002555  012166  104003                        HLT     3               ;COMPARISON ERROR
002556  012170  104400              2$:       SCOPE                   ;SCOPE THIS TEST
002557
002560
002561                                        ;************************ TEST 36 ****************************
002562                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
002563                                        ;*SET BIT10, VERIFY BIT10 WAS SET.
002564                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
002565                                        ;;**********************************************************************
002566
002567                                ;  TEST 36
002570                                ;--------------
002571  012172  012737  000036  001226 TST36:  MOV     #36,TSTNO
002572  012200  012737  012246  001216        MOV     #TST37,NEXT
002573  012206  013703  001372                MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
002574  012212  012705  002000                MOV     #BIT10,R5       ;SET "EXPECTED ".
002575  012216  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
002576  012220  011304                        MOV     (R3),R4         ;READ THE REGISTER.
002577  012222  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
002600  012224  001401                        BEQ     1$              ;ARE THEY THE SAME?
002601  012226  104003                        HLT     3               ;COMPARISON ERROR.
002602  012230  040513              1$:       BIC     R5,(R3)         ;CLEAR BIT10
002603  012232  011304                        MOV     (R3),R4         ;READ THE REGISTER.
002604  012234  005005                        CLR     R5              ;SET "EXPECTED"
002605  012236  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
002606  012240  001401                        BEQ     2$              ;BR IF OK
002607  012242  104003                        HLT     3               ;COMPARISON ERROR
```

```
                        88:     SCOPE                   ;SCOPE THIS TEST

                                ;*********************** TEST 37 ****************************
                                ;*SECONDARY REGISTER SELECTED READ WRITE TEST.
                                ;*PRIMARY DATA ERROR BIT WAS SET.
                                ;*CLEAR DATA, ERROR BIT WAS CLEARED.
                                ;************************************************************

                        ;  TEST 37
                        ;--------------
                        ST37:

                                                        ;SET CSR POINTER INTO R3
                                                        ;SET EXPECTED TO ALL 1'S
                                                        ;WRITE REGISTER WITH EXPECTED DATA
                                                        ;READ THE REGISTER
                                                        ;WAS READ EQUAL TO WRITTEN?
                                                        ;BR IF YES
                                                        ;PRIMARY REGISTER READ WRITE TEST
                        10:                             ;SET DATA TO ZERO
                                                        ;WRITE REGISTER
                                                        ;READ REGISTER
                                                        ;REGISTER WRITTEN OK?
                                                        ;BR IF YES
                                                        ;PRIMARY REGISTER DATA ERROR
                        000:                            ;SET FOR "FLOATING 1"
                        000:                            ;WRITE DATA
                                                        ;READ DATA

                                ;*********************** TEST 40 ****************************
                                ;*SECONDARY REGISTER ACCESS REG. READ WRITE TEST
                                ;*CLEAR BUS INITIALIZE BIT.  ERROR, EACH INDIVIDUAL BIT SET.
                                ;*CLEAR BUS INITIALIZE BIT.  ERROR CLEAR.
                                ;************************************************************

                        ;  TEST 40
                        ;--------------
                        ST40:

                                                        ;SET CSR POINTER INTO R3
                                                        ;SET EXPECTED TO ALL 1'S
                                                        ;WRITE REGISTER WITH EXPECTED DATA
                                                        ;READ THE REGISTER
                                                        ;WAS READ EQUAL TO WRITTEN?
                                                        ;BR IF YES
                                                        ;PRIMARY REGISTER READ WRITE TEST
                        10:                             ;SET DATA TO ZERO
                                                        ;WRITE REGISTER
                                                        ;READ REGISTER
                                                        ;REGISTER WRITTEN OK?
                                                        ;BR IF YES
                                                        ;PRIMARY REGISTER DATA ERROR
                        000:                            ;SET FOR "FLOATING 1"
                        000:                            ;WRITE DATA
                                                        ;READ DATA
```

C05


```
[illegible] 012404 027504            CMP     R5,R4           ;DATA OK?
[illegible]                          BEQ     4$              ;BR IF YES
[illegible]                          [illegible] 3           ;DATA ERROR
[illegible]                   4$:    [illegible]             ;ZERO CPU CARRY
[illegible]                          [illegible] R5          ;UPDATE DATA
[illegible]                          [illegible] 3$          ;BR IF MORE TO GO
[illegible]                          [illegible] (R3)        ;LEAVE REG ALL 0'S
[illegible]                          SCOPE                   ;SCOPE THIS TEST.

                                     ;************************ TEST 41 ******************************
                                     ;*SPECIAL FUNCT. REGISTER READ/WRITE TEST
                                     ;*SET EACH INDIVIDUAL BIT; VERIFY EACH INDIVIDUAL BIT SET.
                                     ;*CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.
                                     ;;*****************************************************************

                              ;       TEST 41
                              ;-----------------
[illegible]                   TST41: [illegible] #41,TSTNO
[illegible]                          [illegible] #TST42,NEXT
[illegible]                          [illegible] #BIT3,ADVSCR   ;SET SOURCE SEL
[illegible]                          [illegible] [illegible],R3    ;SET CSR POINTER INTO R3
[illegible]                          [illegible] #-1,R5      ;SET EXPECTED TO ALL 1'S
[illegible]                          [illegible] R5,(R3)     ;WRITE REGISTER WITH EXPECTED DATA
[illegible]                          [illegible] (R3),R4     ;READ THE REGISTER
[illegible]                          [illegible] R5,R4       ;WAS READ EQUAL TO WRITTEN??
[illegible]                          BEQ     [illegible]     ;BR IF YES
[illegible]                          [illegible]             ;PRIMARY REGISTER READ WRITE TEST [illegible]
[illegible]                   1$:    [illegible] R5          ;SET DATA TO ZERO
[illegible]                          [illegible] R5,(R3)     ;WRITE REGISTER
[illegible]                          [illegible] (R3),R4     ;READ REGISTER
[illegible]                          CMP     R5,R4           ;REGISTER WRITTEN OK?
[illegible]                          [illegible]             ;BR IF YES
[illegible]                          [illegible] 3           ;PRIMARY REGISTER DATA ERROR
[illegible]        000001     2$:    [illegible] #1,R5       ;SET FOR [illegible]
[illegible]                   3$:    [illegible] R5,(R3)     ;WRITE DATA
[illegible]                          [illegible] (R3),R4     ;READ DATA
[illegible]                          CMP     R5,R4           ;DATA OK?
[illegible]                          BEQ     4$              ;BR IF YES
[illegible]                          [illegible] 3           ;DATA ERROR
[illegible]                   4$:    [illegible]             ;ZERO CPU CARRY
[illegible]                          ROL     R5              ;UPDATE DATA
[illegible]                          BNE     3$              ;BR IF MORE TO GO
[illegible]                          CLR     (R3)            ;LEAVE REG ALL 0'S
[illegible]                          SCOPE                   ;SCOPE THIS TEST.

                                     ;************************ TEST 42 ******************************

                                     ;*NPR STATUS REG. TEST
                                     ;*TEST THAT NPR STATUS REG. CANNOT BE WRITTEN
                                     ;*READ THE NPR STATUS REG. AND STORE THE DATA;
                                     ;*COMPLEMENT THE DATA AND  WRITE THE NPR STATUS REG.
                                     ;*VERIFING THAT THE NPR STATUS REG. DID NOT CHANGE.
                                     ;;*****************************************************************
```

D05


```
                                                ;       TEST 42
                                                ;-----------------
[illegible]                                     TST42:  MOV     #42,TSTNO
                                                        MOV     #TST43,NEXT
                                                        MOV     [illegible],R3
                                                                                ;GET THE REGISTER.
                                                        MOV     @R3,R5          ;READ THE REGISTER INTO R5
                                                        MOV     R5,R4           ;SAVE REG INTO R4
                                                        COM     R4              ;R4 OPPOSITE TO REGISTER
                                                        MOV     R4,@R3          ;WRITE THE REGISTER [illegible] COMPLEMENT
                                                        MOV     @R3,R4          ;READ THE REGISTER.
                                                        [illegible] R5,R4       ;[illegible] THE REGISTER EQUAL TO ZERO.
                                                        BEQ     1$              ;BR IF OK.
                                                        [illegible]             ;REGISTER NOT ZERO.
                                                1$:     SCOPE                   ;SCOPE THIS TEST.

                                                        ;************************ TEST 43 ******************************
                                                        ;*DVII RESERVED REGISTER READ WRITE TEST.
                                                        ;*SET BIT0. VERIFY BIT0 WAS SET.
                                                        ;*CLEAR BIT0. VERIFY BIT0 WAS CLEARED.
                                                        ;;**************************************************************

                                                ;       TEST 43
                                                ;-----------------
[illegible]                                     TST43:  MOV     #43,TSTNO
                                                        MOV     #TST44,NEXT
                                                        [illegible]             ;SET REGISTER TO BE TESTED.
                                                        [illegible]             ;SET EXPECTED [illegible]
                                                        [illegible]             ;WRITE THE REGISTER.
                                                        [illegible]             ;READ THE REGISTER.
                                                        [illegible]             ;[illegible] R4=[illegible]
                                                        [illegible]             ;ARE THEY THE SAME?
                                                        [illegible]             ;COMPARISON ERROR.
                                                1$:     [illegible]
                                                        [illegible]             ;READ THE REGISTER.
                                                        [illegible]             ;SET EXPECTED
                                                        [illegible]             ;[illegible] R4=
                                                        [illegible]             ;BR IF OK
                                                        [illegible]             ;COMPARISON ERROR
                                                2$:     SCOPE                   ;SCOPE THIS TEST

                                                        ;************************ TEST 44 ******************************
                                                        ;*DVII RESERVED REGISTER READ WRITE TEST.
                                                        ;*SET BIT1. VERIFY BIT1 WAS SET.
                                                        ;*CLEAR BIT1. VERIFY BIT1 WAS CLEARED.
                                                        ;;**************************************************************

                                                ;       TEST 44
                                                ;-----------------
[illegible]                                     TST44:  MOV     #44,TSTNO
                                                        MOV     #TST45,NEXT
                                                        MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
                                                        MOV     #BIT1,R5        ;SET EXPECTED [illegible]
```

```
                                                        ;WRITE THE REGISTER.
                                                        ;READ THE REGISTER.
                                                        ;R5=GOOD; R4=UNKNOWN.
                                                        ;ARE THEY THE SAME?
                                                        ;COMPARISON ERROR.
                                                        ;CLEAR BIT1
                                                        ;READ THE REGISTER.
                                                        ;SHOULD BE EXPECTED
                                                        ;R5=GOOD; R4=
                                                        ;
                                                        ;COMPARISON ERROR
                                                        ;SCOPE THIS TEST
```

```
;+++++++++++++++++++++++++ TEST 45 ++++++++++++++++++++++++++++++
;+ TEST 45 IS A REGISTER READ/WRITE TEST.
;+ TEST 45 SETS BIT3, VERIFY BIT3 WAS SET.
;+ TEST 45 CLEARS BIT3, VERIFY BIT3 WAS CLEARED.
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
;  TEST 45
;--------------
```

```
                                                        ;SET REGISTER TO BE TESTED.
                                                        ;SET EXPECTED
                                                        ;WRITE THE REGISTER.
                                                        ;READ THE REGISTER.
                                                        ;R5=GOOD; R4=UNKNOWN.
                                                        ;ARE THEY THE SAME?
                                                        ;COMPARISON ERROR.
                                                        ;CLEAR BIT2
                                                        ;READ THE REGISTER.
                                                        ;SHOULD BE EXPECTED
                                                        ;R5=GOOD; R4=
                                                        ;
                                                        ;COMPARISON ERROR
                                                        ;SCOPE THIS TEST
```

```
;+++++++++++++++++++++++++ TEST 46 ++++++++++++++++++++++++++++++
;+ TEST 46 IS A RESERVED REGISTER READ/WRITE TEST.
;+ TEST 46 SETS BIT3, VERIFY BIT3 WAS SET.
;+ TEST 46 CLEARS BIT3, VERIFY BIT3 WAS CLEARED.
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
;  TEST 46
;--------------
```

```
                                                        ;SET REGISTER TO BE TESTED.
                                                        ;SET EXPECTED
                                                        ;WRITE THE REGISTER.
                                                        ;READ THE REGISTER.
                                                        ;R5=GOOD; R4=UNKNOWN.
                                                        ;ARE THEY THE SAME?
```

```
                                                        :COMPARISON ERROR.
                                                        :CLEAR BIT3
                                                        :READ THE REGISTER.
                                                        :SET "EXPECTED"
                                                        :R5=GOOD; R4=?
                                                        :BR IF OK
                                                        :COMPARISON ERROR
                                                        :SCOPE THIS TEST

                                                        ;************************ TEST 47 ******************************
                                                        ;*DZV11 RESERVED REGISTER READ/WRITE TEST.
                                                        ;*SET BIT4, VERIFY BIT4 WAS SET.
                                                        ;*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
                                                        ;;**********************************************************

                                                ;  TEST 47
                                                ;--------------
                                                TST47:  MOV     #47,TSTNO
                                                        MOV     #TST48,.NEXT
                                                        MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
                                                        MOV     #BIT4,R5        ;SET "EXPECTED"
                                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                                        MOV     (R3),R4         ;READ THE REGISTER.
                                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                                        BEQ     1$              ;ARE THEY THE SAME?
                                                        HALT                    ;COMPARISON ERROR.
                                                1$:     BIC     #BIT4,(R3)      ;CLEAR BIT4
                                                        MOV     (R3),R4         ;READ THE REGISTER.
                                                        CLR     R5              ;SET "EXPECTED"
                                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                                        BEQ     2$              ;BR IF OK
                                                        HALT                    ;COMPARISON ERROR
                                                2$:     SCOPE                   ;SCOPE THIS TEST

                                                        ;************************ TEST 50 ******************************
                                                        ;*DZV11 RESERVED REGISTER READ/WRITE TEST.
                                                        ;*SET BITS, VERIFY BITS WAS SET.
                                                        ;*CLEAR BITS, VERIFY BITS WAS CLEARED.
                                                        ;;**********************************************************

                                                ;  TEST 50
                                                ;--------------
                                                TST50:  MOV     #50,TSTNO
                                                        MOV     #TST51,.NEXT
                                                        MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
                                                        MOV     #BITS,R5        ;SET "EXPECTED".
                                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                                        MOV     (R3),R4         ;READ THE REGISTER.
                                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                                        BEQ     1$              ;ARE THEY THE SAME?
                                                        HALT                    ;COMPARISON ERROR.
                                                1$:     BIC     R5,(R3)         ;CLEAR BITS
                                                        MOV     (R3),R4         ;READ THE REGISTER.
                                                        CLR     R5              ;SET "EXPECTED"
```

```
                                                CMP     R5,R4           ;R5=GOOD; R4=?
                                                BEQ     3$              ;OR IF OK
                                                HLT                     ;COMPARISON ERROR
                                        2$:     SCOPE                   ;SCOPE THIS TEST

                                        ;+++++++++++++++++++++++++ TEST 5: ++++++++++++++++++++++++++++++
                                        ;+KV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;+SET BIT 6, VERIFY BIT 6 WAS SET.
                                        ;+CLEAR BIT 6, VERIFY BITS WAS CLEARED.
                                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                        ;  TEST 5:
                                        ;---------------
                                        5$:     [illegible]
                                                [illegible]
                                                [illegible]     ;SET REGISTER TO BE TESTED.
                                                [illegible]     ;SET "EXPECTED"
                                                [illegible]     ;WRITE THE REGISTER.
                                                [illegible]     ;READ THE REGISTER.
                                                [illegible]     ;R5=GOOD; R4=UNKNOWN.
                                                [illegible]     ;ARE THEY THE SAME?
                                                [illegible]     ;COMPARISON ERROR.
                                        10$:    [illegible]     ;CLEAR BIT 6
                                                [illegible]     ;READ THE REGISTER.
                                                [illegible]     ;SET "EXPECTED"
                                                [illegible]     ;R5=GOOD; R4=
                                                [illegible]     ;OR IF OK
                                                [illegible]     ;COMPARISON ERROR
                                        99$:    SCOPE           ;SCOPE THIS TEST

                                        ;+++++++++++++++++++++++ TEST 55 ++++++++++++++++++++++++++++++
                                        ;+KV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;+SET BIT 6, VERIFY BITS WAS SET.
                                        ;+CLEAR BIT 6, VERIFY BITS WAS CLEARED.
                                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                        ;  TEST 55
                                        ;---------------
                                        55$:    [illegible]
                                                [illegible]
                                                [illegible]     ;SET REGISTER TO BE TESTED.
                                                [illegible]     ;SET "EXPECTED"
                                                [illegible]     ;WRITE THE REGISTER.
                                                [illegible]     ;READ THE REGISTER.
                                                [illegible]     ;R5=GOOD; R4=UNKNOWN.
                                                [illegible]     ;ARE THEY THE SAME?
                                                [illegible]     ;COMPARISON ERROR.
                                        60$:    [illegible]     ;CLEAR BIT 6
                                                [illegible]     ;READ THE REGISTER.
                                                [illegible]     ;SET "EXPECTED"
                                                [illegible]     ;R5=GOOD; R4=
                                                [illegible]     ;OR IF OK
                                                [illegible]     ;COMPARISON ERROR
                                        99$:    SCOPE           ;SCOPE THIS TEST
```

```
                                        ;*********************** TEST 54 ****************************
                                        ;*SECONDARY REGISTER READ/WRITE TESTS
                                        ;*READ WRITE TEST. READ AND WRITE DIFFERENT DATA
                                        ;*PATTERENS INTO THE SECONDARY REGISTERS VERIFING
                                        ;*THAT WHAT WAS READ MATCHES WHAT WAS WRITTEN.
                                        ;*************************************************************

                                ;  TEST 54
                                ;--------------
                                T54:    MOV     #T54,@#SWR      ;SET SWR TO TEST NO
                                        MOV     #T54,@#LOOK     ;SET LOOK FOR TEST
                                        CLR     R0              ;SET LINE NUMBER TO ZERO
                                        CLR     R1              ;SET SEC. REG POINTER TO ZERO
                                        MOV     #DVSRA,R2       ;SET ADDRESS POINTER INTO R2
                                1$:     MOV     R0,(R2)         ;LOAD LINE NUMBER
                                        BIS     #DV.SRSH,(R2)   ;LOAD SEC. REG.
                                        CLR     R4              ;ZERO DATA PATTERN
                                2$:     MOV     R4,2(R2)        ;LOAD DATA INTO DVII REGISTER
                                        MOV     2(R2),R3        ;READ BACK THE DATA
                                        CMP     R4,R3           ;WAS WHAT WAS WRITTEN READ BACK??
                                        BEQ     4$              ;BR IF DATA OK
                                6$:     HALT                    ;SECONDARY REGISTER READ WRITE ERROR.
                                        MOV     R4,@#LOOK       ;LOOK ON DATA,LINE,AND SEC REG (SWR9=1
                                        COM     R4              ;MAKE DATA ALL 1'S
                                        MOV     R4,2(R2)        ;RETURN AND WRITE IT INTO THE REGISTER
                                        CMP     R4,R4           ;SET READY FOR THE "FLOATING 1"
                                        BEQ     3$              ;SET IF SWR9=1
                                4$:     MOV     (R2),R4         ;LOAD DATA
                                        MOV     2(R2),R3        ;READ DATA
                                        CMP     R4,R3           ;DATA OK??
                                        BEQ     5$              ;BR IF OK!
                                        HALT                    ;SECONDARY REGISTER READ WRITE ERROR
                                5$:     BCC     1$              ;SWR9=1??
                                        SEC                     ;ZERO CPU CARRY
                                        ROL     R4              ;FLOAT THE 1
                                        BNE     3$              ;IF NOT DONE WRITE NEW PATTERN
                                        MOV     #1,R4           ;GET READY FOR THE "FLOATING 0"
                                3$:     MOV     #DV.SRSH!DV.LOOK,(R2) ;SET IF SWR9=1
                                        MOV     R4,2(R2)        ;WRITE DATA
                                        CMP     (R2),R3         ;READ DATA
                                        CMP     R4,R3           ;DATA OK??
                                        BEQ     6$              ;BR IF OK!
                                        HALT                    ;SECONDARY REGISTER DATA COMPARE ERROR
                                1$:     BCC     1$              ;SWR9=1??
                                        SEC                     ;SET CPU CARRY
                                        ROL     R4              ;CHANGE DATA PATTERN
                                        BCC     2$              ;IF NOT DONE , GO BACK WITH NEW PATTERN
                                        CLR     (R2)            ;ZERO THE REGISTER WE'RE DONE WITH THIS ONE.
                                        INC     R1              ;UPDATE THE SEC REG POINTER
                                        CMP     #16.,R1         ;ALL SEC REGISTERS DONE?
                                        BNE     1$              ;BR IF NO.
                                        CLR     R1              ;ZERO SEC REG POINTER
                                        INC     R0              ;UPDATE LINE NUMBER POINTER
                                        CMP     #16.,R0         ;ALL LINES DONE?
                                        BNE     1$              ;BR IF NO
```

```
013672  104400                  SCOPE                   ;SCOPE THIS TEST


                                ;********************** TEST 55 ****************************
                                ;*INDIVIDUAL LINE DUEL ADDRESS TESTS
                                ;*THIS TEST VERIFIES THAT WRITING ONE SECONDARY
                                ;*REGISTER FOR A SPECIFIC LINE DOES NOT ALTER
                                ;*ANY OTHER SECONDARY REGISTER FOR THAT LINE.
                                ;;**************************************************************

                                ;  TEST 55
                                ;--------------
013674  012737  000055  001226  ^ST55:  MOV     #55,TSTNO
013702  012737  013740  001230          MOV     #TST56,NEXT
013710  012737  013770  001220          MOV     #2$,LOCK
013716  005000                  1$:     CLR     R0                      ;SELECT THE LINE NUMBER.
013720  005001                  65$:    CLR     R1                      ;SET FOR SEC. REG. POINTER.
013722  005004                          CLR     R4                      ;SET DATA
013724  013702  001376                  MOV     @VSRA,R2                ;SET ACCESS REGISTER.
013730  110077  177436                  MOVB    R0,@VSRS                ;SELECT THE LINE NUMBER
013734  110177  177434          2$:     MOVB    R1,@VSRSH               ;SELECT THE SEC. REG.
013740  010412                  3$:     MOV     R4,(R2)                 ;WRITE SEC. REG.
013742  062704  010421                  ADD     #^B0001000010001001,R4  ;UPDATE DATA
013746  005201                          INC     R1                      ;UPDATE SECONDARY REG. POINTER
013750  026127  000020  000020          CMP     #16.,R1                 ;ALL SEC. REG. DONE?
013756  001367                          BNE     3$                      ;BR IF NO
013760  005001                          CLR     R1                      ;RESET SEC. REG. POINTER TO ZERO.
013762  005004                          CLR     R4                      ;ZERO DATA COMPARE
013764  012737  013770  001220          MOV     #2$,LOCK                ;SET FOR LOCK.
013772  110177  177402          2$:     MOVB    R1,@VSRSH               ;GET SEC. REG.
013776  011203                          MOV     (R2),R3                 ;READ SEC. REG.
014000  020403                          CMP     R4,R3                   ;R4=GOOD; R3=UNKNOWN
014002  001401                          BEQ     3$                      ;BR IF ALL OK
014004  104002                          HLT     2                       ;SECONDARY REGISTER ADDRESSING ERROR
014006  104401                  3$:     SCOPE                           ;LOCK ON REG. (SW09=1)
014010  062704  010421                  ADD     #^B0001000010001001,R4  ;UPDATE SEC REG POINTER
014014  005201                          INC     R1                      ;ALL 16 LINES TESTED YET
014016  026127  000020  000020          CMP     #16.,R1                 ;BR IF NO
014024  001363                          BNE     2$                      ;UPDATE LINE NO POINTER
014026  005200                          INC     R0                      ;ALL LINES DONE?
014030  026027  000020  000020          CMP     #16.,R0                 ;BR IF NO
014036  001333                          BNE     65$                     ;SCOPE THE TEST
014040  104400                          SCOPE


                                ;********************** TEST 56 ****************************
                                ;*VERIFY NO LINE INTERACTION.
                                ;*THIS TEST VERIFIES THAT WRITING THE SECONDARY
                                ;*REGISTERS FOR ONE LINE DOES NOT INTERFEAR WITH
                                ;*THE SECONDARY REGISTERS OF ANOTHER LINE.
                                ;;**************************************************************

                                ;  TEST 56
                                ;--------------
014042  012737  000056  001226  ^TST56:  MOV     #56,TSTNO
```

```
                        014042  012737  000002  001332          MOV     #2,ICOUNT
                        014050  012737  014432  001216          MOV     #TST57,NEXT
                        014056  012737  014156  001220          MOV     #4$,LOCK
                        014064  005000                          CLR     R0              ;SELECT THE LINE NUMBER
                        014066  005001                          CLR     R1              ;R1 = SECONDARY REGISTER SELECT.
                        014070  005004                          CLR     R4              ;SET DATA TO BE WRITTEN.
                        014072  013702  001376                  MOV     DVSRA,R2        ;SET "SECONDARY REGISTER ACCESS" POINTER.
                        014076  110077  165266          3$:     MOVB    R0,@DVSRS       ;GET THE LINE NUMBER SELECTED.
                        014102  110177  165266          4$:     MOVB    R1,@DVSRSH      ;GET THE SECONDARY REGISTER.
                        014106  014412                  MOV     R4,(R2)         ;WRITE THE SECONDARY REG.
                        014110  005201                  INC     R1              ;UPDATE  SECONDARY REG. POINTER.
                        014112  022701  000020          CMP     #16.,R1         ;ALL SEC. REG. DONE?
                        014116  001371                  BNE     2$              ;BR IF NO.
                        014120  005001                  CLR     R1              ;ZERO SEC.REG.POINTER
                        014122  062704  010421          ADD     #^B<000100001000010001>,R4
                                                                                ;UPDATE DATA POINTER.
                        014126  005200                  INC     R0              ;UPDATE LINE POINTER.
                        014130  022700  000020          CMP     #16.,R0         ;ALL LINES DONE?
                        014134  001365                  BNE     1$              ;BR IF NO.

                        014136  005000                  CLR     R0              ;SELECT LINE NUMBER
                        014140  005001                  CLR     R1              ;START SEC. REG. AT 0
                        014142  005004                  CLR     R4              ;ZERO DATA COMPARE.
                        014144  110077  165222          MOVB    R0,@DVSRS       ;SELECT LINE NUMBER.
                        014150  110177  165222          MOVB    R1,@DVSRSH      ;SELECT SEC.REG.
                        014154  011203                  MOV     (R2),R3         ;READ THE SEC. REG.
                        014156  020403                  CMP     R4,R3           ;WAS DATA CORRECT?
                        014160  001401                  BEQ     5$              ;BR IF OK
                        014162  104002                  HLT     2               ;SECONDARY REG. ADDRESSING ERROR.
                        014164  104401          5$:     SCOPE                   ;LOCK ON REGISTER? (SW09=1)
                        014166  005201                  INC     R1              ;UPDATE SEC. REG. POINTER
                        014170  022701  000020          CMP     #16.,R1         ;ALL SEC. REG. FOR THIS LINE DONE?
                        014174  001365                  BNE     4$              ;BR IF NO.
                        014176  062704  010421          ADD     #^B<000100001000010001>,R4
                                                                                ;UPDATE DATA
                        014202  005001                  CLR     R1              ;SET FOR SEC. REG.
                        014204  005200                  INC     R0              ;UPDATE LINE NUMBER POINTER.
                        014206  022700  000020          CMP     #16.,R0         ;ALL LINES DONE?
                        014212  001354                  BNE     3$              ;BR IF NO

                                                        ;*PART 2
                                                        ;*FILL ALL RAMS WITH ALL 1'S AND THEN
                                                        ;*CLEAR JUST ONE BIT AT A TIME VERIFYING
                                                        ;*THAT ONLY THAT ONE BIT IS CLEAR AND THAT
                                                        ;*ALL OTHER RAMS STILL CONTAIN ALL 1'S.
                                                        ;*THERE SHOULD BE ONLY ONE BIT CLEARED AT
                                                        ;*ONE TIME.

                        014214  005077  165164          CLR     @RESV16         ;CLEAR "LOC FOUND FLAG".
                        014220  005037  001220          CLR     LOCK            ;
                        014224  013702  001376  MAR17:  MOV     DVSRA,R2        ;LOAD POINTER TO DVSRA
                        014230  005077  165136          CLR     @DVSRS          ;SET LINE AND SEC REG POINTER TO ZERO
                        014234  012712  177777  1$:     MOV     #^B<1111111111111111>,(R2)
                                                                                ;PREPARE TO LOAD ALL 1'S INTO ALL RAM LOC.
                        014240  062777  170361  165124  ADD     #^C BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,@DVSRS
                                                                                ;UPDATE LINE AND SEC REG POINTERS.
```

```
                                                        BNE     1$              ;BR IF NOT ALL ONE FILLING I'S.
                                                        CLR     R0              ;ZERO LINE # POINTER
                                                        CLR     R1              ;ZERO  SEC REG # POINTER
                                        177776  2$:     MOV     #^C^BIT0^,R4    ;SET INITAL DATA
                                                3$:     MOVB    R0,@DVSRS       ;LOAD LINE #
                                                        MOVB    R1,@DVSRSH      ;LOAD SEC REG #
                                                        MOV     R4,(R2)         ;LOAD DATA
                                                        CLR     @DVSRS          ;ZERO POINTERS
                                        165074          CMP     #^B^1111111111111111,(R2)
                                                                                ;VERIFY ONLY ONE LOC. HAS ONE BIT CLEARED
                                                        BEQ     6$              ;BR IF LOC. OK
                                        177777  165072  MOV     #-1,@RESV16     ;SET "LOC FOUND FLAG".
                                                        MOV     (R2),R3         ;SAVE DATA
                                        165052          CMPB    R0,@DVSRS       ;IS THIS THE RIGHT LINE?
                                                        BEQ     4$              ;BR IF YES
                                                        HLT     0               ;WRONG LINE HAS CLEARED BIT!
                                        165044          CMPB    R1,@DVSRSH      ;IS THIS THE RIGHT SEC REG?
                                                4$:     BEQ     5$              ;BR IF YES
                                                        HLT     1               ;WRONG SEC REG HAS CLEARED BIT.
                                                5$:     CMP     R4,R3           ;IS THE ACTUAL DATA OK?
                                                        BEQ     6$              ;BR IF YES
                                                        HLT     2               ;ACTUAL DATA WAS WRONG.
                                        170361  165022  6$:     ADD     #^C^BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0,@DVSRS
                                                        BNE     10$             ;BR IF NOT DONE.
                                        165026          TST     @RESV16         ;HAS A LOC BEEN FOUND?
                                                        BNE     7$              ;BR IF YES
                                                        HLT     0               ;NO LOC WAS FOUND WITH A ZERO BIT.
                                        165016  7$:     CLR     @RESV16         ;CLEAR "LOC FOUND FLAG".
                                                        SEC                     ;SHIFT IN A  1
                                                        ROL     R4              ;CHANGE DATA PATERN
                                                        BCS     3$              ;DO IT ALL OVER AGAIN
                                        164772          MOVB    R0,@DVSRS       ;LOAD LINE NO.
                                        164770          MOVB    R1,@DVSRSH      ;LOAD SEC REG.
                                                        MOV     R4,(R2)         ;PUT RAM LOC BACK TO ALL 1'S.
                                                        INC     R1              ;UPDATE SEC REG #
                                        000020          CMP     #16.,R1         ;ALL SEC REG DONE?
                                                        BNE     2$              ;BR IF NO
                                                        CLR     R1              ;ZERO SEC REG POINTER
                                                        INC     R0              ;UPDATE  LINE POINTER
                                        000020          CMP     #16.,R0         ;ALL LINES DONE?
                                                        BNE     2$              ;BR IF NO
                                                        SCOPE                   ;SCOPE  THIS TEST.


                                                ;************************ TEST 57 ******************************
                                                ;*MEMORY EXTENSION READ/WRITE TEST
                                                ;*VERIFY BITS 4 AND 5 OF EACH LINE
                                                ;*SECONDARY REGISTERS EXERCISED ARE:
                                                ;* 00    TX BUS ADDRESS (PRIMARY)
                                                ;* 02    TX BUS ADDRESS (SECONDARY)
                                                ;* 04    RX BUS ADDRESS
                                                ;* 10    TX TABLE BASE ADDRESS
                                                ;* 11    RX TABLE BASE ADDRESS
                                                ;*NOTE THAT ALL LINES (00-16) ARE EXERCISED.
                                                ;;***********************************************************
```

```
                                        ;  TEST 57
                                        ;--------------
014432  012737  000057  001226  TST57:  MOV     #57,TSTNO
014440  012737  014626  001216          MOV     #TST60,NEXT
014446  012737  014506  001220          MOV     #2$,LOCK
014454  005000                          CLR     R0              ;R0=LINE NUMBER (START AT 0)
014456  013702  001370                  MOV     DVLCR,R2        ;SET R2 =BASE ADDRESS(DVLCR)
014462  012704  000060          1$:     MOV     #BIT5+BIT4,R4   ;R4=GOOD DATA  (BOTH EA BITS SET AT START)
014466  012705  000005                  MOV     #5,R5           ;R5 IS COUNTER OF SEC REGISTERS
014472  012737  000004  001246          MOV     #4,TEMP1        ;TEMP1 IS COUNTER OF COMB. OF EA BITS.
                                                                ;EX: 11,10,01,00
014500  012737  014620  001256          MOV     #MEMEXT,TEMP5   ;TEMP5=SEC. REGISTER POINTER.
014506  010477  164650          2$:     MOV     R4,@DVSCR       ;LOAD DVSCR WITH EA BITS.
014512  110077  164654                  MOVB    R0,@DVSRS       ;SEL THE LINE NUMBER
014516  117701  164534                  MOVB    @TEMP5,R1       ;GET SEC REG.
014522  110177  164546                  MOVB    R1,@DVSRSH      ;SEL THE SEC. REGISTER
014526  005077  164644                  CLR     @DVSRA          ;HIT THE SEC.REG. ACCESS REGISTER.
014532  017703  164532                  MOV     @DVLCR,R3       ;SAVE THE DVLINE PARM. REG.
014536  042703  177717                  BIC     #^C<BIT5+BIT4>,R3
                                                                ;CLEAR ALL BUT BITS 5 AND 4.
014542  020403                          CMP     R4,R3           ;ARE THE EA BITS GOOD
014544  001401                          BEQ     3$
014546  104004                          HLT     4
014550  104401                  3$:     SCOPE1                  ;SW09=1?
014552  005237  001256                  INC     TEMP5           ;POP POINTER
014556  005305                          DEC     R5              ;ALL SEC REG DONE?
014560  001352                          BNE     2$              ;BR IF NO.
014562  012737  014620  001256          MOV     #MEMEXT,TEMP5   ;RESET POINTER
014570  012705  000005                  MOV     #5,R5           ;RESET COUNTER
014574  162704  000020                  SUB     #BIT4,R4        ;ADJUST FOR NEXT EA BIT PATTERN
014600  005337  001246                  DEC     TEMP1           ;ALL PATERNS DONE?
014604  001340                          BNE     2$
014606  005200                          INC     R0              ;UPDATE TO NEXT LINE
014610  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE
014614  001322                          BNE     1$              ;BR IF NO.
014616  104400                          SCOPE
                                ;TABLE OF SECONDARY REGISTERS EXERCISERD....
014620      000                 MEMEXT: .BYTE   00
014621      002                         .BYTE   02
014622      004                         .BYTE   04
014623      010                         .BYTE   10
014624      011                         .BYTE   11
            014626                      .EVEN


                                        ;************************ TEST 60 *******************************
                                        ;*INITIALIZATION TESTS
                                        ;*SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
                                        ;*AND VERIFY THAT ALL THE BITS ARE CLEARED
                                        ;*BY A BUS RESET
                                        ;:*************************************************************


                                        ;  TEST 60
                                        ;--------------
```

```
 3241  014626  012737  000060  001226  TST60:  MOV     #60,TSTNO
 3242  014634  012737  015042  001216          MOV     #TST61,NEXT
 3243  014642  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERUPTS.
 3244  014650  013703  001362                  MOV     DVSCR,R3                ;SET REGISTER POINTER FOR LOADING
 3245  014654  005077  164512                  CLR     @DVSRS                  ;CLEAR LINE POINTER
 3246  014660  005077  164512                  CLR     @DVSRA                  ;CLEAR ACCESS REG.
 3247  014664  012723  173777                  MOV     #^C<BIT11>,(R3)+
 3248  014670  012702  000007                  MOV     #7,R2                   ;SET ALL BITS BUT MSTCLR
 3249  014674  012723  177777          1$:     MOV     #-1,(R3)+               ;LOAD ALL OTHER REGISTERS WITH ALL 1'S
 3250  014700  005302                  DEC     R2                      ;ALLREGISTERS LOADED?
 3251  014702  001374                  BNE     1$                      ;BR IF NO
 3252  014704  000005                  RESET                           ;ISSUES A BUS INIT (RESET INSTR)
 3253  014706  005200                  INC     R0                      ;FLASH THE CPU LIGHTS!!!
 3254  014710  013703  001362          MOV     DVSCR,R3                ;SET REGISTER POINTER
 3255  014714  005005                  CLR     R5                      ;SET "EXPECTED" FOR DVSCR
 3256  014716  011304                  MOV     (R3),R4                 ;READ THE DVSCR REG.
 3257  014720  020504                  CMP     R5,R4                   ;IS BIT8 ALONE SET?
 3258  014722  001401                  BEQ     2$                      ;BR IF YES
 3259  014724  104003                  HLT     3                       ;DVSCR HAS WRONG DATA)
 3260  014726  005723          2$:     TST     (R3)+                   ;POP POINTER TO DVRIC
 3261  014730  005005                  CLR     R5                      ;SET EXPECTED TO ZERO
 3262  014732  011304                  MOV     (R3),R4                 ;DVRIC (EXPECT ALL 0'S)
 3263  014734  001401                  BEQ     3$                      ;BR IF OK
 3264  014736  104003                  HLT     3                       ;DVRIC NO ALL 0'S
 3265  014740  005723          3$:     TST     (R3)+                   ;POP POINTER TO DVLCR REG
 3266  014742  011304                  MOV     (R3),R4                 ;DVLCR (READ DVLCR INTO R4)
 3267  014744  042704  000063          BIC     #BIT5+BIT4+BIT1+BIT0,R4
 3268  014750  020504                  CMP     R5,R4                   ;DISREGUARD BR TEST POINTS AND MEM EXT BITS.
 3269  014752  001401                  BEQ     4$                      ;DVLCR OK?
 3270  014754  104003                  HLT     3                       ;DVLCR INCORECT (DISREGUARD BITS,4,1,0)
 3271  014756  005723          4$:     TST     (R3)+                   ;POP POINTER TO DVSRS REG
 3272  014760  011304                  MOV     (R3),R4                 ;DVSRS (EXPECT ALL 0'S)
 3273  014762  001401                  BEQ     5$                      ;BR IF OK
 3274  014764  104003                  HLT     3                       ;DVSRS REG NOT ALL ZEROS
 3275  014766  005723          5$:     TST     (R3)+                   ;POP POINTER TO DVSRA REG
 3276  014770  011304                  MOV     (R3),R4                 ;DVSRA (EXPECT ALL 0'S)
 3277  014772  001401                  BEQ     6$                      ;BR IF GOOD
 3278  014774  104003                  HLT     3                       ;DVSRA NOT ALL 0'S
 3279  014776  005723          6$:     TST     (R3)+                   ;POP POINTER TO DVSFR
 3280  015000  011304                  MOV     (R3),R4                 ;DVSFR (EXPECT ALL 1'S (THATS RIGHT))
 3281  015002  012705  177777          MOV     #177777,R5              ;SET EXPECTED
 3282  015006  020504                  CMP     R5,R4                   ;EXPECETD =FOUND?
 3283  015010  001401                  BEQ     7$                      ;BR IF YES
 3284  015012  104003                  HLT     3                       ;DVSFR NOT ALL 1'S
 3285  015014  005723          7$:     TST     (R3)+                   ;POP POINTER TO DVNSR REG
 3286  015016  005713                  TST     (R3)                    ;DVNSR S/B PLUS (15=0)
 3287  015020  100001                  BPL     64$
 3288  015022  104000                  HLT     0
 3289  015024  005723          64$:    TST     (R3)+                   ;POP POINTER TO RESV16 REG
 3290  015026  011304                  MOV     (R3),R4                 ;RESV16 (EXPECT ALL 0'S)
 3291  015030  005005                  CLR     R5                      ;SET EXPECTED TO 0'S
 3292  015032  020504                  CMP     R5,R4                   ;WELL DOES IT =1'S?
 3293  015034  001401                  BEQ     8$                      ;BR IF OK
 3294  015036  104003                  HLT     3                       ;RESV16 NOT ALL 0'S
 3295  015040  104400          8$:     SCOPE                           ;SCOPE THIS TEST:
 3296
```

```
                                        ;********************** TEST 61 ****************************
                                        ;*INITIALIZATION TESTS
                                        ;*SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
                                        ;*AND VERIFY THAT ALL THE BITS ARE CLEARED
                                        ;*BY A MASTER CLEAR
                                        ;*************************************************************

                                ;  TEST 61
                                ;-------------
```

```
                                                        ;POP POINTER TO RES/16 REG
                                                        ;REG.16 (EXPECT ALL 0'S)
                                                        ;SET EXPECTED TO 0'S
                                                        ;DOES IT =1'S?
                                                        ;BR IF OK
                                                        ;RES/16 NOT ALL 0'S
                                                        ;SCOPE THIS TEST;

;********************** TEST 62 ****************************
;ATTACH OF THE SPECIAL FUNCTIONS REGISTER.
;BEGIN CHECK OF THE DVSFR.
;SUMMARY OF PROC. INSTRUCTIONS
;
;BIT15 BIT13 BIT12  INSTRUCTION
;  0     0     0    BRANCH "B"
;  0     0     1    ALU OPERATION
;  0     1     0    ARM OPERATION
;  0     1     1    DATA TRANSFER
;  1     0     0    SPR OPERATION
;  1     0     1    MISCELLANEOUS CLEAR OPERATION
;  1     1     0    MISCELLANEOUS CALCULATION
;  1     1     1    RESERVED "B"
;
;*************************************************************

;********************** TEST 62 ****************************
;VERIFY THAT "ROM STEP"
;IS SELF-CLEARING AND THAT
;THE DATA IN THE DVSFR
;IS CHANGED WHEN THE ROM IS STEPPED.
;*************************************************************

;  TEST 62
;---------------
TSTE62: MOV     #62,TSTNO
        MOV     #TS%63,NEXT
        CLR                             ;CLEAR ALL THE BITS
        MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
        MOV     #SCC.RS,.BUT INSTR INTO DVSFR
        MOV     R5,@DVSFR
        MOV     @DVSFR,R5               ;WAS THE DVSFR REALLY LOADED?
        BR      1$                      ;BR IF YES
                                        ;BAD DVSFR
        MOV     #BIT3,@DVSCR            ;CLEAR SOURCE SEL.
        CLR                             ;ISSUE A ROM CLOCK
                                        ;WAIT AN INSTRUCTION TIME
        BIT     #BIT1,@DVSCR            ;DIO CLR BIT CLEAR BY ROM SELF?
        BEQ     2$                      ;BR IF CLR GONE
                                        ;BIT 1 OF DVSCR (ROM CLR) NOT ZERO
        MOV     @DVSFR,R5               ;HAS DATA IN DVSFR CHANGED?
        BNE     3$                      ;BR IF YES
                                        ;DATA NOT CHANGED (DIO CLR FAILED CLEAR)
3$:     SCOPE                           ;SCOPE THIS TEST.
```

```
                        ;********************** TEST 63 ******************************
                        ;*BASIC TEST OF DVSFR
                        ;*TEST THAT "BRANCH A" INSTRUCTION.
                        ;*POINTS TESTED:
                        ;*
                        ;*BIT:  BIT10  BIT09  BIT08  BP "A"  BP "B"  FUNCTION
                        ;*  0     0      0      0      1       0      [illegible]
                        ;*  0     1      1      1      1       0      [illegible]
                        ;*  0     1      1      0      1       0      [illegible]
                        ;*  0     1      1      1      0       1      [illegible]
                        ;*
                        ;;************************************************************

                        ; TEST 63
                        ;-----------
[illegible]  000063:    [illegible]     #63,ISTNO
                        [illegible]     #BTST,NEXT
                        [illegible]CLR
                        [illegible]     DVSFR,R0        ;CLEAR DVSFR
                        [illegible]     DVLOR,R0        ;SET DVSFR POINTER TO 00
                        [illegible]     #BIT10+BIT09+BIT08,@DVSFR ;SET DVLOR POINTER TO 07
[illegible]  163744     [illegible]18:  #BIT10+BIT09,@DVSFR ;BR-A TEST +3
                        [illegible]     @DVSFR,R2       ;READ DVSFR FOR PRINTOUT
                        [illegible]     #BIT10,R5       ;SET EXPECTED RESULTS
                        [illegible]     DVLOR,R4        ;READ DVLOR INTO R4
                        [illegible]     #BIT10+BIT09,R4
                                                        ;CLEAR UNWANTED BITS.
                        [illegible]CMP  R5,R4           ;EXPECTED = FOUND?
                        [illegible]     BEQ .+6         ;BR IF OK
                        [illegible]     JMP @FAIL       ;BR POINT WRONG
[illegible]  000400     [illegible]88:  #BIT10+BIT9,@DVSFR ;BR-A TEST +2
[illegible]  000000     [illegible]     @DVSFR,R2       ;READ DVSFR FOR PRINTOUT
[illegible]  177774     [illegible]     #BIT10,R5       ;SET EXPECTED RESULTS
                        [illegible]     DVLOR,R4        ;READ DVLOR INTO R4
                        [illegible]     #BIT10+BIT9,R4
                                                        ;CLEAR UNWANTED BITS.
                        [illegible]CMP  R5,R4           ;EXPECTED = FOUND?
                        [illegible]     BEQ .+6         ;BR IF OK
                        [illegible]     JMP @FAIL       ;BR POINT WRONG
[illegible]  000400     [illegible]88:  #BIT10+BIT9,@DVSFR ;BR-A TEST +1
[illegible]  000000     [illegible]     @DVSFR,R2       ;READ DVSFR FOR PRINTOUT
[illegible]  177774     [illegible]     #BIT10,R5       ;SET EXPECTED RESULTS
                        [illegible]     DVLOR,R4        ;READ DVLOR INTO R4
                        [illegible]     #BIT10+BIT9,R4
                                                        ;CLEAR UNWANTED BITS.
                        [illegible]CMP  R5,R4           ;EXPECTED = FOUND?
                        [illegible]     BEQ .+6         ;BR IF OK
                        [illegible]     JMP @FAIL       ;BR POINT WRONG
[illegible]  000400     [illegible]88:  #BIT10+BIT9+BIT8,@DVSFR ;BR-A TEST +0
[illegible]  000000     [illegible]     @DVSFR,R2       ;READ DVSFR FOR PRINTOUT
[illegible]  177774     [illegible]     #BIT10,R5       ;SET EXPECTED RESULTS
                        [illegible]     DVLOR,R4        ;READ DVLOR INTO R4
                        [illegible]     #BIT10+BIT9,R4
                                                        ;CLEAR UNWANTED BITS.
                        [illegible]CMP  R5,R4           ;EXPECTED = FOUND?
```

```
                                                BNE     5$              ;BR IF OK
                                                HALT                    ;BR POINT WRONG
                                                MOV     #BIT11+BIT10+BIT9+BIT8,(R0)
                                5$:             MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                                                MOV     #BIT1+BITC,R5   ;SET EXPECTED RESULTS
                                                MOV     (R3),R4         ;READ DVLCR INTO R4
                                                BIC     #^C<BIT1+BITC>,R4
                                                                        ;CLEAR UNWANTED BITS.
                                                CMP     R5,R4           ;EXPECTED = FOUND
                                                BEQ     6$              ;BR IF OK
                                                HALT                    ;BR POINT WRONG
                                6$:             MOV     #BIT10+BIT9,(R0)
                                                MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                                                MOV     #BIT1+BITC,R5   ;SET EXPECTED RESULTS
                                                MOV     (R3),R4         ;READ DVLCR INTO R4
                                                BIC     #^C<BIT1+BITC>,R4
                                                                        ;CLEAR UNWANTED BITS.
                                                CMP     R5,R4           ;EXPECTED = FOUND
                                                BEQ     7$              ;BR IF OK
                                                HALT                    ;BR POINT WRONG
                                7$:             SCOPE                   ;SCOPE TEST

                                        ;********************** TEST 64 ****************************
                                        ;*TEST OF BRANCH 8"
                                        ;*TEST THAT POINT IS "GROUND"
                                        ;*MAKES DCR BIT1=1 AND BIT0=1.
                                        ;;*********************************************************

                                ;  TEST 64
                                ;--------------
                                ST64:   MOV     #64,TSTNO
                                        MOV     #TST65,NEXT
                                        MTPS    #CLR                    ;CLEAR DV11
                                        MOV     #BIT3,DCVSCR            ;SET SOURCE SEL
                                        MOV     #BRB+BIT11+BIT10+BIT9+BIT8,DCVSFR
                                                                        ;BR-B "GROUND"
                                        MOV     DCVSFR,R2               ;READ DVSFR INTO R2
                                        MOV     #BIT1+BITC,R5           ;SET EXPECTED RESULTS
                                        MOV     DCVLCR,R4               ;READ REAL RESULTS
                                        CMP     R5,R4                   ;SAME?
                                        BEQ     13$                     ;
                                        HALT                            ;BR TEST POINT WRONG
                                13$:    SCOPE                           ;SCOPE THIS TEST

                                        ;********************** TEST 65 ****************************
                                        ;*TEST OF BRANCH B
                                        ;*CHECKING "DEFAULT" STATES OF THE DV11 SIGNALS.
                                        ;*BIT11 BIT10   BIT09   BIT08   FUNCTION
                                        ;*  1     0       0       0     DATA NOT AVAIL.
                                        ;*  1     0       0       1     REQUEST B.S
                                        ;*  1     0       1       0     MEMORY PARITY ERROR
                                        ;*  1     1       1       1     WRITE INHIBIT
                                        ;;*********************************************************

                                ;  TEST 65
                                ;--------------
```

```
                                                TST65:  MOV     #65,TSTNO
                                                        MOV     #TST66,.NEXT
                                                        MSTCLR
                                                        MOV     DVSFR,R0
                                                        MOV     DVLCR,R3
                                                        MOV     #BIT3,@DVSCR
                                                1$:     MOV     #BRB+BIT11,(R0)
                                                        MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                                                        MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                                                        MOV     (R3),R4         ;READ DVLCR INTO R4
                                                        BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                                                        CMP     R5,R4           ;EXPECTED = FOUND??
                                                        BEQ     2$              ;BR IF OK
                                                        6                       ;BR POINT WRONG
                                                2$:     MOV     #BRB+BIT11+BIT9.(R0)
                                                        MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                                                        MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                                                        MOV     (R3),R4         ;READ DVLCR INTO R4
                                                        BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                                                        CMP     R5,R4           ;EXPECTED = FOUND??
                                                        BEQ     3$              ;BR IF OK
                                                        6                       ;BR POINT WRONG
                                                3$:     MOV     #BRB+BIT11+BIT9.(R0)
                                                        MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                                                        MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                                                        MOV     (R3),R4         ;READ DVLCR INTO R4
                                                        BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                                                        CMP     R5,R4           ;EXPECTED = FOUND??
                                                        BEQ     4$              ;BR IF OK
                                                        6                       ;BR POINT WRONG
                                                4$:     MOV     #BRB+BIT11+BIT10+BIT9.(R0)
                                                        MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                                                        MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                                                        MOV     (R3),R4         ;READ DVLCR INTO R4
                                                        BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                                                        CMP     R5,R4           ;EXPECTED = FOUND??
                                                        BEQ     5$              ;BR IF OK
                                                        6                       ;BR POINT WRONG
                                                5$:     SCOPE


                                                ;********************** TEST 66 ****************************
                                                ;*BASIC TEST OF THE
                                                ;*"SET/CLEAR INSTRUCTION.
                                                ;*TEST THAT THE SET/CLEAR CAN DO:
                                                ;*CLEAR DVSCR 08
                                                ;*SET DVSCR10
                                                ;*SET RECEIVER INTERUPT (DVSCR07)
                                                ;;**********************************************************

                                                ;  TEST 66
```

```
                                                ;--------------
                                        TST66:  MOV     #66,TSTNO
                                                MOV     #TST6,.NEXT
                                                MOV     #1$,LOCK
                                                MSTCLR                  ;CLEAR DV11
                                                BIS     #BIT8,@DVSCR    ;SET BIT8.
                                                BIS     #BIT3,@DVSCR    ;SET SOURCE SEL.
                                                MOV     DVSFR,R0        ;SET DVSFR POINTER ADDRESS IN R0
                                        1$:     MOV     #5,0+2IT3+BIT2+BIT1,(R0)
                                                MOV     #BIT3,R5        ;DO SET/CLEAR -CLEAR BIT 9 OF DVSCR
                                                MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                ROMCLK                  ;CYCLE THE ROM
                                                MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
                                                CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
                                                BEQ     64$             ;BR IF DVSCR OK
                                                HLT     6               ;ROM FAILED TO EXECUTE
                                        64$:    SCOPE                   ;LOCK ON THIS SUB-TEST? SW09=1?
                                                MOV     #3$,LOCK        ;SET FOR RETURN IF SW09=1
                                                BIC     #BIT4,R5        ;SET EXPECTED DVSCR BIT 7=1
                                        3$:     MOV     #5,0+BIT3+BIT1+BIT0,(R0)
                                                MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                ROMCLK                  ;CYCLE THE ROM
                                                MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
                                                CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
                                                BEQ     65$             ;BR IF DVSCR OK
                                                HLT     6               ;ROM FAILED TO EXECUTE
                                        65$:    SCOPE                   ;LOCK ON THIS SUB-TEST? SW09=1?
                                                MOV     #4$,LOCK
                                                BIC     #BIT10,R5       ;ALTER EXPECTED ADDRESS
                                        4$:     MOV     #5,0+BIT3+BIT2+BIT1,(R0)
                                                                        ;DO A SET/CLEAR SET DVSCR BIT 10
                                                MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                ROMCLK                  ;CYCLE THE ROM
                                                MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
                                                CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
                                                BEQ     66$             ;BR IF DVSCR OK
                                                HLT     6               ;ROM FAILED TO EXECUTE
                                        66$:    SCOPE                   ;LOCK ON THIS SUB-TEST? SW09=1?
                                                SCOPE                   ;SCOPE THIS TEST


                                                ;************************ TEST 67 ******************************
                                                ;*BASIC TEST OF THE
                                                ;*"SET/CLEAR INSTRUCTION.
                                                ;*TEST THAT THE SET/CLEAR CAN:
                                                ;*
                                                ;*BIT 14  BIT12  BIT03  BIT02  BIT01  BIT00  FUNCTION
                                                ;*  1      1      1      0      0      1      SET BIT00
                                                ;*  1      1      1      0      1      0      SET BIT01
                                                ;*  1      1      1      1      0      0      SET BIT02
                                                ;*  1      1      1      1      1      0      SET BIT03
                                                ;*
                                                ;;**********************************************************

                                        ;  TEST 67
                                        ;--------------
```

```
                                                TSTST:  MOV     #ST.TSTNO
                                                        BIS     #ST.NEXT
                                                        CLR     DVSCR           ;CLEAR DVII
                                                        MOV     #DVSFR,RO       ;SET DVSFR POINTER TO RO
                                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                        MOV     #BIT15,R5       ;SET EXPECTED RESULTS
                                                        MOV     #BIT15+BIT3,(RO)
                                                                                ;SET/CLEAR DVRICR 15
                                                        MOV     (RO),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                RDMCLK: MOV     @DVRIC,R4       ;CYCLE THE ROM
                                                        CMP     R4,R4           ;READ DVRIC INTO "FOUND" LOC.
                                                        BEQ     .+4             ;WAS THE ROM INSTR EXECUTED
                                                        TST     (PC)+           ;BR IF DVSCR OK
                                                        SCOPE                   ;ROM FAILED TO EXECUTE
                                                        BR      RDMCLK          ;LOOK ON THIS SUB-TEST" SWCB=1
                                                        CLR     DVSCR           ;CLEAR DVII
                                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                        MOV     #BIT12,R5       ;SET EXPECTED RESULTS
                                                        MOV     #BIT15+BIT12+BIT3,(RO)
                                                                                ;SET/CLEAR DVRICR 14
                                                        MOV     (RO),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                RDMCLK: MOV     @DVRIC,R4       ;CYCLE THE ROM
                                                        CMP     R4,R4           ;READ DVRIC INTO "FOUND" LOC.
                                                        BEQ     .+4             ;WAS THE ROM INSTR EXECUTED
                                                        TST     (PC)+           ;BR IF DVSCR OK
                                                        SCOPE                   ;ROM FAILED TO EXECUTE
                                                        BR      RDMCLK          ;LOOK ON THIS SUB-TEST" SWCB=1
                                                        CLR     DVSCR           ;CLEAR DVII
                                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                        MOV     #BIT13,R5       ;SET EXPECTED RESULTS
                                                        MOV     #BIT15+BIT13+BIT3,(RO)
                                                                                ;SET/CLEAR DVRICR 13
                                                        MOV     (RO),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                RDMCLK: MOV     @DVRIC,R4       ;CYCLE THE ROM
                                                        CMP     R4,R4           ;READ DVRIC INTO "FOUND" LOC.
                                                        BEQ     .+4             ;WAS THE ROM INSTR EXECUTED
                                                        TST     (PC)+           ;BR IF DVSCR OK
                                                        SCOPE                   ;ROM FAILED TO EXECUTE
                                                        BR      RDMCLK          ;LOOK ON THIS SUB-TEST" SWCB=1
                                                        CLR     DVSCR           ;CLEAR DVII
                                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                        MOV     #BIT12,R5       ;SET EXPECTED RESULTS
                                                        MOV     #BIT15+BIT13+BIT12+BIT3+BIT2+BIT0,(RO)
                                                                                ;SET/CLEAR DVRICR 12
                                                        MOV     (RO),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
                                                RDMCLK: MOV     @DVRIC,R4       ;CYCLE THE ROM
                                                        CMP     R4,R4           ;READ DVRIC INTO "FOUND" LOC.
                                                        BEQ     .+4             ;WAS THE ROM INSTR EXECUTED
                                                        TST     (PC)+           ;BR IF DVSCR OK
                                                        SCOPE                   ;ROM FAILED TO EXECUTE
                                                        BR      RDMCLK          ;LOOK ON THIS SUB-TEST" SWCB=1
                                                        SCOPE                   ;SCOPE THIS TEST

                                                ;********************** TEST 70 **************************
                                                ;*BASIC TEST OF DVSFR.
```

```
                                        ;*TEST OF "SET/CLEAR" AND
                                        ;*"BRANCH 4" AND "BRANCH 8" FUNCTIONS.
                                        ;;**************************************************************

                                ;  TEST 70
                                ;-------------
016600  012700  000070  001000  TST70:  MOV     #70,TSTNO
016604  012701  016744  001004          MOV     #SSRTN,NEXT
016610  012702  016644  001010          MOV     #10$,LOOK
016614  005000                          CLR     @LOOK                   ;CLEAR DVII
016616  012703  000010  177600          MOV     #BIT3,@DVSCR            ;SET SOURCE OF ECT
016622  012704  016744                  MOV     DVSPR,R0                ;SET DVSPR POINTER INTO R0

                                        ;*TEST SET CLEAR FUNCTION
                                        ;*FOR ALL OUTPUTS
016626  004737  017134          10$:    JSR     R2,10$                  ;GOTO SUBROUTINE
016632  000024                          BIT4+BIT2               ;POINT SET/CLEARED
016634  000011                          BIT11+BIT10             ;BR TEST POINT
016636  000001                          BIT1                    ;EXPECTED RESULTS IN DVLOR
016640  004737  017134                  JSR     R2,10$          ;GOSUB
016644  000024                          BIT4+BIT2+BIT0          ;POINT SET CLEARED
016646  000011                          BIT11+BIT10             ;BR TEST POINT
016650  000001                          BIT1+BIT0               ;EXPECTED RESUTS IN DVLOR
016652  104401                          SCOP1                   ;SWR09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT00
016654  012737  016660  001220          MOV     #2$,LOCK                ;SET RETURN IF SW09=1
016662  004737  017134          2$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016666  000047                          BIT5+BIT2+BIT1+BIT0                     ;POINT SET CLEARED (SET)
016670  000000                          BRB                     ;BR TEST POINT
016672  000000                          BIT0                    ;DVLOR EXPECTED
016674  004737  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016700  000046                          BIT5+BIT1+BIT0                  ;POINT SET CLEARED (CLEARED)
016702  000000                          BRB                     ;BR TEST POINT
016704  000000                          BIT1+BIT0               ;EXPECTED RESULTS
016706  104401                  65$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT01
016710  012737  016634  001220          MOV     #3$,LOCK                ;SET RETURN IF SW09=1
016716  004737  017134          3$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016722  000045                          BIT5+BIT2+BIT0                  ;POINT SET CLEARED (SET)
016724  000000                          BRB+BIT8                ;BR TEST POINT
016726  000001                          BIT0                    ;DVLOR EXPECTED
016730  004737  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016734  000045                          BIT5+BIT0               ;POINT SET/CLEARED (CLEARED)
016736  000000                          BRB+BIT8                ;BR TEST POINT
016740  000003                          BIT1+BIT0               ;EXPECTED RESULTS
016742  104401                  67$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT02
016744  012737  016670  001220          MOV     #4$,LOCK                ;SET RETURN IF SW09=1
016752  004737  017134          4$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016756  000046                          BIT5+BIT2+BIT1                  ;POINT SET CLEARED (SET)
```

```
                                                BNB+BIT9                        ;BR TEST POINT
                                                BIT0                    ;DVLCR EXPECTED
                                                JSR     R2,10$          ;GOTO SUB ROUTINE
                                                BIT5+BIT1                       ;POINT SET/CLEARED (CLEARED)
                                                BRB+BIT9                        ;BR TEST POINT
                                                BIT1+BIT0               ;EXPECTED RESULTS
                                        69$:    SCOP1                   ;SWR 09=1?

                                                ;*TEST SET CLEAR FUNCTION
                                                ;*FOR RAM OUTPUT BIT03
                                                MOV     #6$,LOCK                ;SET RETURN IF SW09=1
                                        6$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
                                                BIT5+BIT2                       ;POINT SET CLEARED (SET)
                                                BRB+BIT9+BIT8                   ;BR TEST POINT
                                                BIT0                    ;DVLCR EXPECTED
                                                JSR     R2,10$          ;GOTO SUB ROUTINE
                                                BIT5                    ;POINT SET/CLEARED (CLEARED)
                                                BRB+BIT9+BIT3                   ;BR TEST POINT
                                                BIT1+BIT0               ;EXPECTED RESULTS
                                        71$:    SCOP1                   ;SWR 09=1?

                                                ;*TEST SET CLEAR FUNCTION
                                                ;*FOR RAM OUTPUT BIT04
                                                MOV     #7$,LOCK                ;SET RETURN IF SW09=1
                                        7$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
                                                BIT7+BIT2+BIT1+BIT0                     ;POINT SET/CLEARED (SET)
                                                BRB+BIT10                       ;BR TEST POINT
                                                BIT0                    ;DVLCR EXPECTED
                                                JSR     R2,10$          ;GOTO SUB ROUTINE
                                                BIT7+BIT1+BIT0                  ;POINT SET/CLEARED (CLEARED)
                                                BRB+BIT10                       ;BR TEST POINT
                                                BIT1+BIT0               ;EXPECTED RESULTS
                                        73$:    SCOP1                   ;SWR 09=1?

                                                ;*TEST SET/CLEAR FUNCTION
                                                ;*FOR RAM OUTPUT BIT05
                                                MOV     #8$,LOCK                ;SET RETURN IF SW09=1
                                        8$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
                                                BIT7+BIT2+BIT0                  ;POINT SET/CLEARED (SET)
                                                BRB+BIT10+BIT8                  ;BR TEST POINT
                                                BIT0                    ;DVLCR EXPECTED
                                                JSR     R2,10$          ;GOTO SUB ROUTINE
                                                BIT7+BIT0                       ;POINT SET/CLEARED (CLEARED)
                                                BRB+BIT10+BIT8                  ;BR TEST POINT
                                                BIT1+BIT0               ;EXPECTED RESULTS
                                        75$:    SCOP1                   ;SWR 09=1?

                                                ;*TEST SET/CLEAR FUNCTION
                                                ;*FOR RAM OUTPUT BIT06
                                                MOV     #9$,LOCK                ;SET RETURN IF SW09=1
                                        9$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
                                                BIT7+BIT2+BIT1                  ;POINT SET CLEARED (SET)
                                                BRB+BIT10+BIT9                  ;BR TEST POINT
                                                BIT0                    ;DVLCR EXPECTED
                                                JSR     R2,10$          ;GOTO SUB ROUTINE
                                                BIT7+BIT1                       ;POINT SET CLEARED (CLEARED)
```

```
                                                BNB+BIT10+BIT9                  ;BR TEST POINT
                                                BIT1+BIT0                       ;EXPECTED RESULTS
                                        77$:    SCOPE1                          ;SWR 09=1?

                                                :*TEST SET CLEAR FUNCTION
                                                :*FOR RAM OUTPUT BIT07
                                                MOV     #101$,LOCK              ;SET RETURN IF SW09=1
                                        101$:   JSR     R2,10$  ;GOTO THE SUBROUTINE
                                                BIT7+BIT2                       ;POINT SET/CLEARED (SET)
                                                BRB+BIT10+BIT9+BIT8             ;BR TEST POINT
                                                BIT0                            ;DVLCR EXPECTED
                                                JSR     R2,10$                  ;GOTO SUB ROUTINE
                                                BIT7                            ;POINT SET/CLEARED (CLEARED)
                                                BRB+BIT10+BIT9+BIT8             ;BR TEST POINT
                                                BIT1+BIT0                       ;EXPECTED RESULTS
                                        79$:    SCOPE1                          ;SWR 09=1?
                                                SCOPE                           ;SCOPE THE TEST
                                        10$:    MOV     #S.C,(R0)               ;SET/CLEAR INSTR
                                                MOV     R0,R1                   ;SAVE JSR PC ADDRESS
                                                BIS     (R2)+,(R0)              ;LOAD POINT SET/CLEARED
                                                ROMCLK                          ;CYCLE THE ROM
                                                MOV     (R2)+,(R0)              ;LOAD BR TEST POINT
                                                MOV     (R0),R3                 ;READ DVSFR INTO R3
                                                MOV     (R2)+,R5                ;LOAD EXPECTED INTO R5
                                                MOV     @DVLCR,R4               ;READ DVLCR INTO FOUND LOC.
                                                CMP     R5,R4                   ;EXPECTED=FOUND?
                                                BEQ     11$                     ;BR IF YES
                                                HLT     5                       ;DVLCR WRONG BR RESULTS.
                                        11$:    RTS     R2                      ;RETURN

                                                ;********************** TEST 71 ****************************
                                                :*TEST OF "RECEIVER CHARACTER SILO"
                                                :*THRU THE USE OF THE DVSFR REG.
                                                :*TEST THE FILLING THE SILO PRODUCES "SILO FULL"
                                                :*ON EXACTLY THE 128 LOAD.
                                                :*SET/CLEAR IS USED TO STUFF SILO AND BRANCH A IS USED TO TEST SILO.
                                                :*SET/CLEAR "SILO IN" AND SET/CLEAR "SILO OUT" ARE EXERCISED TOO.
                                                ;:***********************************************************

                                                :  TEST 71
                                                :--------------
017170  012737  000071  001226  TST71:  MOV     #71,TSTNO
017176  012737  017412  001216          MOV     #TST72,NEXT
017204  104412                          MSTCLR                          ;CLEAR DV11
017206  012700  000177                  MOV     #127.,R0                ;SET R0 TO 1 LESS THAN FULL SILO
017212  012777  000010  162142          MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
017220  012705  000003                  MOV     #BIT1+BIT0,R5           ;SET EXPECTED RESULTS INTO R5
017224  012777  050021  162146  15$:    MOV     #S.C+BIT4+BIT0,@DVSFR
017232  104415                          ROMCLK                          ;S/C "SILO IN"
017234  012777  007400  162136          MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                                        ;BR-A "SILO FULL"?
017242  017702  162132                  MOV     @DVSFR,R2               ;SAVE CONTENTS OF DVSFR FOR ERROR PRINTOUT
017246  017704  162116                  MOV     @DVLCR,R4               ;READ DVLCR FOR RESULTS
017252  020504                          CMP     R5,R4                   ;ARE BR TEST POINTS CORRECT?
017254  001401                          BEQ     64$                     ;BR IF YES
017256  104036                          HLT     6                       ;BR TEST POINTS WRONG (BIT1 OR 0
```

```
017260  005300                  64$:    DEC     R0              ;IS SILO FULL-1 YET
017262  001360                          BNE     1$              ;BR IF NOT 127 TIMES YET
017264  012777  050021  162106    2$:   MOV     #S.C+BIT4+BIT0,@DVSFR
                                        ;S/C "SILO IN"
017272  104415                          ROMCLK                  ;
017274  000240                          NOP                     ;WAIST INSTRJCTION TIME
017276  012777  007400  162074          MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                        ;BR-A "SILO FULL"?
017304  017702  162070                  MOV     @DVSFR,R2       ;SAVE DVSFR
017310  017704  162054                  MOV     @DVLCR,R4       ;READ BR TEST POINTS
017314  042705  000001                  BIC     #BIT0,R5        ;ALTER EXPECTED RESULTS
017320  020504                          CMP     R5,R4           ;BR TEST POINTS OK??
017322  001401                          BEQ     3$              ;BR IF YES
017324  104006                          HLT     6               ;BR TEST POINTS WRONG
017326  012777  050020  162044    3$:   MOV     #S.C+BIT4,@DVSFR
017334  104415                          ROMCLK                  ;S/C "SILO OUT"
017336  000240                          NOP                     ;WAIST INSTR TIME
017340  012777  007400  162032          MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                        ;BR-A "SILO FULL"?
017346  005002                          CLR     R2              ;DEALAY AT LEAST 32US
017350  032777  000001  162012    4$:   BIT     #BIT0,@DVLCR    ;IS SILO *NOT FULL*?
017356  001003                          BNE     5$              ;BR IF OK.
017360  062702  000001                  ADD     #1,R2           ;DELAY.....
017364  001371                          BNE     4$              ;GOTO 4$
017366  017702  162006            5$:   MOV     @DVSFR,R2       ;SAVE DVSFR
017372  017704  161772                  MOV     @DVLCR,R4       ;READ BR TEST POINTS
017376  052705  000001                  BIS     #BIT0,R5        ;SET EXPECTED RESULTS
017402  020504                          CMP     R5,R4           ;OK??
017404  001401                          BEQ     6$              ;YES
017406  104006                          HLT     6               ;SILO STILL FULL.
017410  104400                  6$:     SCOPE                   ;SCOPE TEST

                                        ;********************** TEST 72 ****************************
                                        ;*TEST THAT AFTER AN INIT
                                        ;*THAT "RECV CHARACTER WAITING"
                                        ;*IS FALSE (HIGH) AND THEN VERIFY
                                        ;*THAT WHEN "SILO IN" IS ASSERTED THAT
                                        ;*THAT "RCVED CHARACTER WAITING" IS TRUE (LOW)
                                        ;*AND MAKES "BRANCH A" TRUE.
                                        ;:**********************************************************

                                        ;  TEST 72
                                        ;--------------
017412  012737  000072  001226  TST72:  MOV     #72,TSTNO
017420  012737  017616  001216          MOV     #TST73,NEXT
017426  104412                          MSTCLR                  ;CLEAR DV11
017430  012777  000010  161724          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
017436  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
017442  012702  001400                  MOV     #BIT9+BIT8,R2   ;BR-A "RCVD CHAR WAITING"?
017446  010277  161726                  MOV     R2,@DVSFR       ;LOAD DV INSTR
017452  017704  161712                  MOV     @DVLCR,R4       ;READ TEST POINTS
017456  020504                          CMP     R5,R4           ;OK??
017460  001401                          BEQ     64$             ;YES
017462  104006                          HLT     6               ;TEST POINT RECV CHAR WAITING WRONG
017464  012702  050021          64$:    MOV     #S.C+BIT4+BIT0,R2
                                        ;S/C "SILO IN"
```

```
3913  017470  010277  161704                  MOV     R2,@DVSFR       ;LOAD INSTR
3914  017474  104415                          ROMCLK                  ;CLOCK
3915  017476  005004                          CLR     R4              ;PREPARE COUNTER
3916  017500  012702  001400                  MOV     #BIT9+BIT8,R2   ;BR-A RCV CHAR WAITING
3917                                                                  ;BR-A "RCVD CHAR WAITING"?
3918  017504  010277  161670                  MOV     R2,@DVSFR       ;LOAD INSTR
3919  017510  012705  000002                  MOV     #BIT1,R5        ;SET GOOD RESULTS
3920  017514  032777  000001  161646  1$:     BIT     #BIT0,@DVLCR    ;TEST DV11 BR POINT
3921  017522  001403                          BEQ     2$              ;BR IF OK
3922  017524  062704  000001                  ADD     #1,R4           ;DELAY
3923  017530  001371                          BNE     1$              ;GOTO 1$
3924  017532  017704  161632          2$:     MOV     @DVLCR,R4       ;READ DV11 BR POINT
3925  017536  020504                          CMP     R5,R4           ;
3926  017540  001401                          BEQ     3$
3927  017542  104006                          HLT     6               ;BR POINT RCV CHAR WAITING WRONG
3928                                          ;*TEST THAT SETTING DVSCR07
3929                                          ;*INHIBITS RCV CHAR WAITING FROM APPEARING
3930                                          ;*TRUE; AND THAT CLEARING
3931                                          ;*DVSCR07 MAKES IT APPEAR TRUE AGAIN.
3932
3933  017544  012705  000003          3$:     MOV     #BIT1+BIT0,R5   ;LOAD EXPECTED
3934  017550  052777  001200  161604          BIS     #BIT9+BIT7,@DVSCR       ;SET RECV INTER
3935  017556  017704  161606                  MOV     @DVLCR,R4       ;READ DV BR POINTS
3936  017562  020504                          CMP     R5,R4           ;
3937  017564  001401                          BEQ     4$              ;
3938  017566  104006                          HLT     6               ;BR TEST POINTS WRONG
3939  017570  042705  000001          4$:     BIC     #BIT0,R5        ;RESET EXPECTED RESULTS
3940  017574  042777  000200  1:              BIC     #BIT7,@DVSCR    ;CLEAR RECV INT
3941  017602  017704  161562                  MOV     @DVLCR,R4       ;READ BR POINTS
3942  017606  020504                          CMP     R5,R4           ;
3943  017610  001401                          BEQ     5$              ;
3944  017612  104006                          HLT     6               ;BR TEST POINTS WRONG
3945  017614  104400          5$:             SCOPE                   ;SCOPE THIS TEST
3946
3947
3948                                          ;************************ TEST 73 *******************************
3949                                          ;*BASIC TEST OF THE "DATA TRANSFER INSTRUCTION"
3950                                          ;*BITS 07,06,05,04 OF DVSFR INDICATE THE SOURCE
3951                                          ;*BITS 03,02,01,00 OF DVSFR INDICATE THE DESTINATION.
3952                                          ;;*****************************************************************
3953
3954                                  ;  TEST 73
3955                                  ;--------------
3956  017616  012737  000073  001226  TST73:  MOV     #73,TSTNO
3957  017624  012737  020156  001216          MOV     #TST74,NEXT
3958  017632  012737  017732  001220          MOV     #1$,LOOK
3959  017640  104412                          MSTCLR                  ;CLEAR DV11
3960  017642  012777  000010  161512          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
3961  017650  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER INTO R0
3962
3963                                          ;*TEST TO XFR SOURCE REGISTERS TO THE DVRIC
3964                                          ;*REGISTER VERIFYING THAT THE FOLLOWING REGISTERS
3965                                          ;*ARE CLEARED AND THAT THE XFR BUS IS CLEAR AFTER
3966                                          ;*A MSTCLR.
3967                                          ;*REGISTER      FUNCTION
3968                                          ;* 0000          GROUND
```

```
                                                        ;* 0001          GROUND
                                                        ;* 0010          GROUND
                                                        ;* 0011          GROUND
                                                        ;* 0100          GROUND
                                                        ;* 0101          MASTER SCAN 0-3/0-3
                                                        ;* 0110          ALU RESULT 8-11/0-3
                                                        ;* 0111          ALU RESULT 5-7/0-2
                                                        ;* 1000          LOW BYTE=B REG 8-15 ; HIGH BYTE=GRND
                                                        ;* 1001          LO BYTE=NPR OUT ; HI BYTE=CCC REG
                                                        ;* 1010          RAM OUTPUT 0-2/8-10
                                                        ;* 1011          RAM OUTPUT
                                                        ;* 1101          NPR INPUT REGISTER
                                                        ;* 1110          BCC REGISTER
                                                        ;* 1111          ALU RESULT REGISTER

 017654  005005                                         CLR     R5              ;SET EXPECTED TO 0
 017656  012702  030000                                 MOV     #XFR,R2         ;SET DATA XFR INSTR.
 017662  052702  000038                                 BIS     #BIT2+BIT1,R2   ;SET DESTINATION TO DVRIC REG.
 017666  005003                                         CLR     R3              ;ZERO SOURCE REG POINTER
 017670  042702  000360          65$:                   BIC     #BIT7+BIT6+BIT5+BIT4,R2
 017674  050302                                         BIS     R3,R2           ;SET SOURCE REGISTER
 017676  010210                                         MOV     R2,(R0) ;LOAD SFR WITH XFR INSTR
 017700  104415                                         ROMCLK                  ;EXECUTE INSTR
 017702  017704  161460                                 MOV     @DVRIC,R4       ;READ SOURCE REGISTER
 017706  001401                                         BEQ     66$             ;BR IF IT WAS ZERO
 017710  104006                                         HLT     6               ;SOURCE REGISTER IN SFR NOT ZERO
 017712  062703  000020          66$:                   ADD     #BIT4,R3        ;UPDATE SOURCE REGISTER
 017716  023703  000300                                 CMP     #300,R3         ;DON'T DO SILO REGISTER!!
 017722  001773                                         BEQ     66$             ;GET NEXT REG IF THIS IS SILO.
 017724  032703  000360                                 BIT     #BIT7+BIT6+BIT5+BIT4,R3
 017730  001357                                         BNE     65$             ;BR IF MORE TO DO.

                                                        ;*TEST OF SET RAM OUTPUT BIT0
                                                        ;*AND THE USE OF THE DATA XFER INSTR.
                                                        ;*PLACE RAM BIT0 INTO THE DVRIC REG
 017732  012705  000400          1$:                    MOV     #BIT8,R5
 017736  012703  030000                                 MOV     #XFR,R3 ;-DATA XFER-
 017742  052703  000246                                 BIS     #BIT7+BIT5+BIT2+BIT1,R3
 017746  004237  020076                                 JSR     R2,10$          ;S= RAM OUTPUT 0-2. D= DVRIC
 017752  000047                                         BIT5+BIT2+BIT1+BIT0
 017754  000043                                         BIT5+BIT1+BIT0

                                                        ;*TEST TO SET RAM OUTPUT DATA BIT3
                                                        ;*AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                                                        ;*TO PLACE BIT3 INTO THE DVRIC REGISTER.
 017756  012737  017764  001220                         MOV     #3$,LOCK        ;SET RETURN IF SW09=1
 017764  012705  000010          3$:                    MOV     #BIT3,R5        ;SET EXPECTED DATA
 017770  012703  030000                                 MOV     #XFR,R3 ;-DATA XFER-
 017774  052703  000268                                 BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
 020000  004237  020076                                 JSR     R2,10$          ;S= RAM OUTPUT. D=DVRIC
 020004  000044                                         BIT5+BIT2
 020006  000040                                         BIT5

                                                        ;*TEST TO SET RAM OUTPUT DATA BIT4
                                                        ;*AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                                                        ;*TO PLACE BIT4 INTO THE DVRIC REGISTER.
```

```
                                                        ;SET RETURN IF ERROR=1
                                                        ;SET EXPECTED DATA
                                                        ;DATA XFER-
                                                        ;SR RAM OUTPUT. D=D.FIC
                                                        ;TEST TO SET RAM OUTPUT DATA BIT7
                                                        ;AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                                                        ;TO PLACE BIT7 INTO THE D.FIC REGISTER.
                                                        ;SET RETURN IF ERROR=1
                                                        ;SET EXPECTED DATA
                                                        ;DATA XFER-
                                                        ;SR RAM OUTPUT. D=D.FIC
                                                        ;SET CLEAR POINT
                                                        ;LOAD SET CLEAR POINT
                                                        ;EXECUTE
                                                        ;LOAD XFER
                                                        ;EXECUTE
                                                        ;READ RESULTS
                                                        ;OK??
                                                        ;YES
                                                        ;XFER FAILED
                                                        ;SET CLEAR
                                                        ;POINT
                                                        ;EXECUTE
                                                        ;XFER
                                                        ;EXECUTE
                                                        ;READ EXPECTED RESULTS
                                                        ;READ RESULTS
                                                        ;RESULTS OK
                                                        ;EXIT SUB
;********************** TEST 74 ****************************
;*BASIC TEST OF THE "ALU OPERATION" INSTRUCTION.
;*FIRST PART:ISSUE AN INIT AND MOVE
;*THE ALU RESULT REGISTER TO THE DVRIC
;*REGISTER VERIFING THAT IT IS ZERO.
;*SECOND PART: DO A FUNCTION "F=A"
;*THEN MOV "F" TO THE D.RIC REGISTER VERIFING IT TO
;*BE ZERO
;*THIRD PART: DO A FUNCTION "F=A+B", MOVING
;*"F" TO DVRIC AND MAKING SURE IT IS ZERO.
;*THUS THE FOLLOWING HAS BEEN TESTED:
;*ALU RESULT,"A" REG.AND "B" REG ALL ZEROED ON INIT.
;***********************************************************

: TEST 74
```

```
                                        ;--------------
                                        ;TST4:  MOV     #T4,TSTNO
                                                MOV     #TS+TS.NEXT
                                                CLR                     ;RESET DV11
                                                MOV     #BIT3.OR.SCR    ;SET SOURCE SELECT
                                                MOV     DVSFR,RO        ;SET DVSFR POINTER TO RO
                                                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,(RO)
                                                                        ;XFR "ALU RESULT REG." TO DATA
                                                MOV     #1,CLOCK        ;CLOCK INSTRUCTION
                                                MOV     #0,R0           ;READ "EXPECTED" LOC
                                                CMP     R0,R1           ;READ "ALU RESULT"
                                                BEQ     .+6             ;S B=0
                                                JSR     R5,ERROR        ;ALU RESULT NOT=0 ON INIT
                                                .WORD   ^B<ALU+BIT4+BIT3+BIT2+BIT1+BIT0>,R0
                                                                        ;DO "ALU F=A"
                                                MOV     (R0),DVSFR      ;XFR "ALU RESULT" TO DATA
                                                MOV     #1,CLOCK        ;CLOCK INSTR
                                                CMP     R0,R1,R4        ;READ RESULT
                                                BEQ     .+6             ;S B=0
                                                JSR     R5,ERROR        ;"A" REG NOT=0 ON INIT
                                                .WORD   ^B<ALU+BIT4+BIT2+BIT0>  ;(1) PC)
                                                MOV     #1,CLOCK        ;ALU F=A+B
                                                MOV     R0,DVSFR        ;XFR TO R1
                                                MOV     #1,CLOCK        ;CLOCK INSTR.
                                                CMP     R0,R1,R4        ;READ RESULT
                                                BEQ     .+6             ;S B=0
                                                JSR     R5,ERROR        ;"B" REG NOT=0 ON INIT
                                                .WORD   0
                                        ;T4:    SCOPE                   ;SCOPE TEST

                                        ;********************** TEST 75 ****************************
                                        ;*TEST OF ALU OPERATIONS
                                        ;*TEST OF ALL ALU OPERATIONS USED BY DV11.
                                        ;*FUNCTIONS TESTED: NOTE THAT "F" IS ALU RESULT
                                        ;*DVSFR BITS:
                                        ;*BIT12 BIT05 BIT04 BIT03 BIT02 BIT01 BIT00 FUNCTION
                                        ;*   1     0     0     1     0     1     1    F=-1
                                        ;*   1     0     0     1     1     0     0    F=0
                                        ;*   1     0     0     1     0     1     0    F=A
                                        ;*   1     0     0     1     1     1     1    F=A+1
                                        ;*   1     0     0     1     0     0     1    F=A+B
                                        ;**********************************************************

                                        ;  TEST 75
                                        ;--------------
                                        ;ST75:  MOV     #T5,TSTNO
                                                MOV     #TS+TS.NEXT
                                                ;*FUNCTION TESTED
                                                ;*F=-1,R1C=F
                                                ;*
                                                MOV     #100.,LOCK      ;SET FOR SW09
                                        ;100:   MSTCLR                  ;CLEAR DV11
                                                MOV     #BIT3.OR.SCR,(R0) ;SET SOURCE SEL
                                                MOV     DVSFR,R0        ;SET DVSFR POINTER TO RO
                                                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
                                                MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
```

```
                                                    ;DO "F=-1"
                        R2, PD                      ;XFR "ALU RESULT TO D.RIC"
                                                    ;
                        PD,RIC,R4                   ;READ RESULTS
                        #-1,R5                      ;SET EXPECTED
                        R4                          ;DID F=-1 WORK?
                        6$                          ;BR IF YES
                        6                           ;F=-1 APPEARED TO FAIL
;TEST THAT D.RIC, NOW THAT ITS ALL 1'S
;CAN BE CLEARED BY A MISCLR.
;
CF1:                                                ;SWC9=1?
                        #11$,LOOP                   ;SET RETURN IF SWC9=1
                        MISCLR                      ;CLEAR D.RIC
                        R5                          ;SET EXPECTED RESULTS
                        PD,RIC,R4                   ;READ D.RIC
                        #0                          ;DID IT=0
                        6                           ;D.RIC REG. NOT CLEARED ON MISC
;NEXT SET OF FUNCTIONS:
;F=D.RIC+F
;
                        #BIT3,SD,SCR                ;SET SOURCE SEL
                        #ALU+BIT4+BIT3+BIT2,PD      ;F=-1
                        #ALU+BIT3+BIT2,PD           ;F=0
                        R2,(PD)                     ;XFR "F" TO D.RIC
                                                    ;
                        R5                          ;SET EXPECTED
                        PD,RIC,R4                   ;READ RESULTS
                        6$                          ;BR IF =0
                        6                           ;"F=0" APPEARED TO FAIL
;NEXT SET OF FUNCTIONS:
;F=R+1,RIC=F,A=F,F=0,F=A
;
CF1:                                                ;SWC9=1?
                        #12$,LOOP                   ;SET RETURN
                        #1,R5                       ;SET GOOD RESULTS
                        #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,RO
                        #BIT1,SD,SCR                ;ISSUE ROM CLK
                        RO,(RO)                     ;XFR "F" TO D.RIC
                        #BIT1,SD,SCR                ;ISSUE ROM CLK
                        D.RIC,R4                    ;READ RESULTS
                        R5,R4                       ;FUNCTION WORK?
                        6$                          ;YES
                        6                           ;F=R+1 APPEARED TO FAIL
                        #XFR+BIT7+BIT5+BIT5+BIT4+BIT0,RO
                        #BIT1,SDVSCR                ;ISSUE ROM CLK
                        #ALU+BIT3+BIT2,RO)
                        #BIT1,SDVSCR                ;ROM CLK
                        R2,(PD)                     ;XFR RIC=F
                        #BIT1,SDVSCR                ;ROM CLK
                        D.RIC,R4                    ;READ RESULT
                        R5,R4                       ;BR IF GOOD
                        6$                          ;F=0 FAILED
                        6
                        #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,RO
```

E07


```
                                                        ;CLOCK "F=A"
                                                        ;XFR RIC←F
                                                        ;ROM CLK
                                                        ;READ RESULT
                                                        ;BR IF OK
                                                        ;
                                                        ;F=A FAILED
                                                        ;UPDATE CNTA
                                                        ;BR IF NOT ALL DONE.
                                                        ;SLOW=1?
                                                        ;
                                                        ;NEXT SET OF FUNCTIONS:
                                                        ;F=A←1.B←F.S←F.F=A←9.RIC←F
                                                        ;
                                                        ;SET RETURN
                                                        ;RESET CNTL
                                                        ;INIT SOURCE SEL.
                                                        ;INIT EXPECTED RESULTS
                                                        ;INIT POINTER
                                                        ;INIT PATTERN
                                                        ;ALU←BIT5←BIT4←BIT3←BIT2←BIT1←BIT0.←RC.
                                                        ;F=A←1
                                                        ;XFR←BIT7←BIT6←BIT5←BIT4←BIT0.←RC.
                                                        ;XFR A←ALU RESULT
                                                        ;XFR←BIT7←BIT6←BIT5←BIT4←BIT1.←RC.
                                                        ;XFR S←ALU RESULT
                                                        ;←BIT3←BIT2←BIT1.←RC.
                                                        ;CLOCK F=A←9
                                                        ;XFR RIC←F
                                                        ;
                                                        ;READ CNTRIC
                                                        ;FUNCTION WORK?
                                                        ;WORK←F (YES)
                                                        ;←B EXPECTED TO FAIL
                                                        ;INC DATA POINTER
                                                        ;
                                                        ;
                                                        ;
                                                        ;
                                                        ;NEXT SET OF FUNCTIONS:
                                                        ;F=-1.B←F.F=0.F=B
                                                        ;
                                                        ;SET RETURN
                                                        ;RESET CNTL
                                                        ;INIT SOURCE SEL.
                                                        ;ALU←BIT4←BIT3←BIT2.←RC)
                                                        ;F=-1
                                                        ;XFR←BIT7←BIT6←BIT5←BIT4←BIT1.←RC.
                                                        ;XFR S←F
                                                        ;←BIT3←BIT2.←RC) :
                                                        ;F=0
                                                        ;←BIT2←BIT0.←RC) :
                                                        ;F=0
                                                        ;XFR RIC←F
```

```
                                                                    PAGE 84
                                                                    COPYRIGHT 1975  DIGITAL EQUIP. CORP.

                                                                    ;*****
                                                                    ;*THIS PRO
                                                                    ;*NEXT EXPE
                                                                    ;*OID F=B L
                                                                    ;*OR IF YES
                                                                    ;*F=B FAILE
                                                                    ;SWD9=1
                                                                    SCOPE
                                                                    ;*NEXT SET OF FUNCTIONS:
                                                                    ;*CATCH "SET CLEAR" THAT WAS MISSED.
                                                                    ;*#-1.S CIALU0I=01,RIC+F
                                                                    ;*
                                                                    #15$.LOCK       ;SET RETURN
                                                                    CLR             ;RESET DVII
                                                                    #BIT3,ADVSCR    ;SET SOURCE SELECT
                                                                    #ALU+BIT4+BIT3+B.  2, RO)
                                                                    CLK             ;F=-1
                                                                    #5.0+BIT4+BIT2+B  1,(R0)
                                                                    CLK             ;S/A "SET CLEAR ALU0I"
                                                                    R2,(R0)         ;XFR R1C+F
                                                                    CLK             ;
                                                                    (R1),R4         ;READ DVRIC
                                                                    #1777775,R5     ;SET EXPECTED
                                                                    R5,R4           ;OID S/C ALU0I WORK
                                                                    BEQ 1$          ;BR IF YES
                                                                    6               ;S/C ALU0I FAILED.
                                                                    SCOPE           ;SWD9=1
                                                                    ;*NEXT SET OF FUNCTIONS:
                                                                    ;*CATCH ANOTHER "SET CLEAR" THAT WAS MISSED.
                                                                    ;*#-1.S CIALU HIGH BYTE=01,RIC+F
                                                                    ;*
                                                                    #16$.LOCK       ;SET RETURN
                                                                    CLR             ;RESET DVII
                                                                    #BIT3,ADVSCR    ;SET SOURCE SEL
                                                                    #ALU+BIT4+BIT3+BIT2,(R0)
                                                                    CLK             ;F=-1
                                                                    #5.0+BIT4+BIT2+BIT1+BIT0,(R0)
                                                                    CLK             ;S/A "CLEAR ALU HIGH BYTE"
                                                                    R2,(R0)         ;XFR R1C+F
                                                                    CLK             ;
                                                                    (R1),R4         ;READ RIC
                                                                    #177777,R5      ;SET EXPECTED
                                                                    R5,R4           ;OID S/C WORK
                                                                    BEQ 1$          ;BR IF YES
                                                                    6               ;S/C ALU HIGH BYTE FAILED.
                                                                    SCOPE           ;SWD9=1
                                                                    ;SCOPE TEST

                                                                    ;************************ TEST 75 ******************************
                                                                    ;*MASTER SCANNER TEST.
                                                                    ;*VERIFY FIRST THAT THE MASTER SCANNER
                                                                    ;*IS CLEARED BY INIT.
                                                                    ;*VERIFY SECONDLY THAT THE MASTER SCANNER
                                                                    ;*CAN BE INCREMENTED FROM 0 THRU 17 BACK TO 0.
                                                                    ;***********************************************************
```

```
                                        ;  TEST 76
                                        ;-------------
000000  000000  000000  000000  000000  $T76:  MOV     #76,TSTNO
000002  000132  000132  000132  000132         MOV     #$T4+2,NEXT
000006  000000  000000  000000  000000         MTCLR                   ;RESET DV11
000010  000000  000000  160130  160130         MTPS    #BIT3.OR.SCR    ;SET SOURCE SEL
000014  000000  000000  000000  000000         MOV     #XFR,R0         ;SET DVSFR POINTER
000020  000000  000000  000000  000000         MTPS    #XFR+BIT6+BIT4+BIT3+BIT1,R2
000024  000000  000000  000000  000000         R2, R0                  ;XFR "MASTER SCAN 0-3" TO D.RIC
000030  000000  000000  000000  000000         ROMCLK                  ;
000034  000000  000000  000000  000000         MFPS    R0              ;SET EXPECTED
000040  000000  000000  160134  160134         MOV     #$T4,RIC.R4     ;READ RESULTS
000044  000000  000000  000000  000000         BIT     #8,R0           ;BIT TF=0
000050  000000  000000  000000  000000         ;MSCAN NOT=0 ON INIT
000050  000000  000000  000000  000000         ;UPDATE POINTER
000050  000000  000000  000000  000000  1$:    MOV     #25.,R3         ;SET COUNT TO 25
000054  000000  000000  000000  000000  2$:    BIS     #BIT5+BIT3+BIT1,R0      ;S C "ADVANCE MSCAN"
000060  000000  000000  000000  000000         ;CLOCK
000060  000000  000000  000000  000000         BIC     #BIT5+BIT3+BIT1,R0      ;S C "ADVANCE MSCAN"
000064  000000  000000  000000  000000         ;CLOCK
000064  000000  000000  000000  000000         R2,R0                   ;XFR RIC=MSCAN
000070  000000  000000  000000  000000         ;CLOCK
000070  000000  000000  000000  000000         ROMCLK                  ;CLOCK
000074  000000  000000  160056  160056         MOV     RIC.R4,R4       ;READ RESULTS
000100  000000  000000  000000  000000         ;MSCAN INCREMENTED
000100  000000  000000  000000  000000         ;BIT TF YES
000100  000000  000000  000000  000000         ;MSCAN WRONG
000100  000000  000000  000000  000000  3$:    ;UPDATE
000100  000000  000000  000000  000000         SOB     R3,2$           ;LOOP
000100  000000  000000  000000  000000         ;MSCAN CNT DONE
000100  000000  000000  000000  000000         ;BIT TF NO
000100  000000  000000  000000  000000         ;SCOPE

                                        ;+++++++++++++++++++++++ TEST 77 +++++++++++++++++++++++++++++
                                        ;+BASIC TESTS OF THE "RAM OPERATION" INSTRUCTION.
                                        ;+VERIFY THE READ PORTION OF THE RAM OPERATION.
                                        ;+LOAD ALL SECONDARY REGISTERS OF ALL LINES
                                        ;+WITH DIFFERENT NUMBERS AND VERIFY THAT THE RAM OPERATION
                                        ;+CAN READ THE CORRECT SEC. REG. INTO THE DATA REG.
                                        ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                        ;  TEST 77
                                        ;-------------
000102  000000  000000  000000  000000  $T77:  MOV     #77,TSTNO
000106  000000  000000  000000  000000         MOV     #$T5+2,NEXT
000112  000000  000000  000000  000000         MTCLR                   ;RESET DV11
000116  000000  000000  000000  000000         CLR     R0              ;
000120  000000  000000  000000  000000         CLR     R1              ;
000122  000000  000000  000000  000000         MOV     #XMSCAN,R0      ;
000126  000000  000000  000000  000000         MOV     #BRSCN,R1       ;
000132  000000  000000  000000  000000         MOV     #3,R5           ;
000136  000000  000000  000000  000000         MOV     #-1,R4          ;
000142  000000  000000  000000  000000  1$:    MTPS    R0              ;LOAD LINE NUMBER
000144  000000  000000  000000  000000         MTPS    R1              ;LOAD SEC. REG. POINTER
000146  000000  000000  000000  000000         MTPS    R4              ;LOAD DATA
000150  000000  000000  000000  000000         CMPB    R0,R4           ;UPDATE LINE AND DATA
000152  000000  000000  000000  000000         CMP     #10.,R0         ;ALL LINES DONE
```

```
                                                        ;BR IF NO
                                                        ;CLEAR LINK POINTER
                                                        ;UPDATE SEQ REG POINTER.
                                                        ;ALL SEQ REG POINTERS DONE?
                                                        ;BR IF NO
                                                        ;CLEAR POINTERS
                                                        ;SET GOOD DATA
                                                        ;SET SOURCE SEL
                                                        ;LOAD RAM INSTR.
                                                        ;SET POINTER
                                                        ;DO RAM READ
                                                        ;EXECUTE
                                                        ;XFR RIO<-RAM OUTPUT
                                                        ;CLOCK
                                                        ;READ RESULT
                                                        ;GOOD?
                                                        ;BR IF YES
                                                        ;RAM READ FAILED.
                                                        ;UPDATE DATA
                                                        ;UPDATE POINTER
                                                        ;ALL DONE
                                                        ;BR IF NO
                                                        ;CLEAR JUNK
                                                        ;RAM INSTR
                                                        ;SAVE R0 ON STACK
                                                        ;UPDATE MSCANNER
                                                        ;
                                                        ;RESTORE R0
                                                        ;UPDATE DATA
                                                        ;XFR RAR-MSCAN 0-3
                                                        ;EXECUTE
                                                        ;ALL DONE?
                                                        ;BR IF NO
                                                        ;SCOPE TEST.

                                                        ;********************* TEST 100 ****************************
                                                        ;*TEST OF BRANCH A TEST POINTS
                                                        ;*THAT WERE PREVIOUSLY SKIPPED BECAUSE
                                                        ;OF THE SIGNALS NEEDED TO TEST THE
                                                        ;*FUNCTIONS:
                                                        ;*BIT11 BIT10   BIT09   BIT08   FUNCTION
                                                        ;*  0     0       0       0     ALU (A+1)
                                                        ;*  0     0       0       1     ALU (A-1)
                                                        ;*  0     0       1       0     ALU (A+2)
                                                        ;*  0     0       1       1     ALU (-1)
                                                        ;*  0     1       0       0     ALU (A+B)
                                                        ;*  0     1       0       1     ALU (A-B)
                                                        ;*  0     1       1       0     ALU (A OR B)
                                                        ;*  0     1       1       1     ALU (A AND B)
                                                        ;*********************************************************

                        ; TEST 100
                        ;-----------
000000 016700 000100 001026 TST100: MOV     #100,TSTNO
000006 016700 000100 001216         MOV     #TST101,NEXT
```

```
                                                CLR     @DVSR5
                                                ;*BRANCH "9" TEST OF ALU 15
                                                ;*
                                                MSTCLR                  ;RESET DVII
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                MOV     #BIT15,@DVSRA   ;LOAD DATA
                                                MOV     #RAM,@DVSFR     ;DO A "RAM READ"
                                                ROMCLK                  ;EXECUTE
                                                MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
                                                ROMCLK                  ;XFR RAM OUTPUT TO "9" REG
                                                MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
                                                ROMCLK                  ;F=A
                                                MOV     #0000,@DVSFR    ;LOAD BRANCH POINT
                                                MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
                                                BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
                                                MOV     #BIT1,R5        ;SET EXPECTED
                                                CMP     R5,R4           ;BR POINTS CORRECT?
                                                BEQ     64$             ;BR IF YES
                                                HLT     6               ;BRANCH POINTS WRONG
                                        64$:    MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
                                                MOV     #0000,@DVSFR    ;RESET DVII
                                                MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
                                                BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
                                                MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
                                                CMP     R5,R4           ;SET EXPECTED
                                                BEQ     65$             ;BR POINT OK?
                                                HLT     6               ;BR IF YES
                                        65$:                            ;BR POINTS WRONG
                                                ;*BRANCH "A" TEST OF ALU 13-
                                                ;*
                                                MSTCLR                  ;RESET DVII
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                MOV     #BIT13,@DVSRA   ;LOAD DATA
                                                MOV     #RAM,@DVSFR     ;DO A "RAM READ"
                                                ROMCLK                  ;EXECUTE
                                                MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
                                                ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
                                                MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
                                                ROMCLK                  ;F=A
                                                MOV     #BIT11,@DVSFR   ;LOAD BRANCH POINT
                                                MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
                                                BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
                                                MOV     #BIT1,R5        ;SET EXPECTED
                                                CMP     R5,R4           ;BR POINTS CORRECT?
                                                BEQ     66$             ;BR IF YES
                                                HLT     6               ;BRANCH POINTS WRONG
                                        66$:    MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
                                                MOV     #BIT11,@DVSFR   ;RESET DVII
                                                MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
                                                BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
                                                MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
                                                CMP     R5,R4           ;SET EXPECTED
                                                BEQ     67$             ;BR POINT OK?
                                                HLT     6               ;BR IF YES
```

```
                                                57$:                            ;BR POINTS WRONG
                                                                :*BRANCH "A" TEST OF ALU  -12
                                                                :*
022074  104412                                                MSTCLR                          ;RESET DV11
022076  012777  000010  157256                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
022104  012777  010000  157264                                MOV     #BIT12,@DVSRA   ;LOAD DATA
022112  012777  020000  157260                                MOV     #RAM,@DVSFR     ;DO A "RAM READ"
022120  104415                                                ROMCLK                          ;EXECUTE
022122  012777  030261  157250                                MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022130  104415                                                ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
022132  012777  010037  157240                                MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022140  104415                                                ROMCLK                          ;F=A
022142  012777  004000  157230                                MOV     #BIT11,@DVSFR   ;LOAD BRANCH POINT
022150  017704  157214                                        MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022154  042704  177774                                        BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022160  012705  000002                                        MOV     #BIT1,R5        ;SET EXPECTED
022164  020504                                                CMP     R5,R4           ;BR POINTS CORRECT?
022166  001401                                                BEQ     59$             ;BR IF YES
022170  104006                                                HLT     6               ;BRANCH POINTS WRONG
022172  104412                                        59$:    MSTCLR
022174  012777  000010  157160                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022202  012777  004000  157170                                MOV     #BIT11,@DVSFR   ;RESET DV11
022210  017704  157154                                        MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
022214  042704  177774                                        BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022220  012705  000003                                        MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
022224  020504                                                CMP     R5,R4           ;SET EXPECTED
022226  001401                                                BEQ     69$             ;BR POINT OK?
022230  104006                                                HLT     6               ;BR IF YES
022232                                                  69$:                            ;BR POINTS WRONG
                                                                :*BRANCH "A" TEST OF ALU 00
                                                                :*
022232  104412                                                MSTCLR                          ;RESET DV11
022234  012777  000010  157120                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
022242  012777  000001  157126                                MOV     #BIT0,@DVSRA    ;LOAD DATA
022250  012777  020000  157122                                MOV     #RAM,@DVSFR     ;DO A "RAM READ"
022256  104415                                                ROMCLK                          ;EXECUTE
022260  012777  030261  157112                                MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022266  104415                                                ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
022270  012777  010037  157102                                MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022276  104415                                                ROMCLK                          ;F=A
022300  012777  005000  157072                                MOV     #BIT11+BIT9,@DVSFR      ;LOAD BRANCH POINT
022306  017704  157056                                        MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022312  042704  177774                                        BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022316  012705  000002                                        MOV     #BIT1,R5        ;SET EXPECTED
022322  020504                                                CMP     R5,R4           ;BR POINTS CORRECT?
022324  001401                                                BEQ     70$             ;BR IF YES
022326  104006                                                HLT     6               ;BRANCH POINTS WRONG
022330  104412                                        70$:    MSTCLR
022332  012777  000010  157022                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022340  012777  005000  157032                                MOV     #BIT11+BIT9,@DVSFR      ;RESET DV11
022346  017704  157016                                        MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
022352  042704  177774                                        BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022356  012705  000003                                        MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
022362  020504                                                CMP     R5,R4           ;SET EXPECTED
022364  001401                                                BEQ     71$             ;BR POINT OK?
022366  104006                                                HLT     6               ;BR IF YES
```

```
                                                    ;BP POINTS WRONG
                        ;*BRANCH "9" TEST OF ALU 01
                        ;*
022370  104412          MSTCLR                      ;RESET DV11
022372  012777  000010  156762      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
022400  012777  000002  156756      MOV     #BIT1,@DVSRA    ;LOAD DATA
022406  012777  020000  156764      MOV     #RAM,@DVSFR     ;DO A "RAM READ"
022414  104415          ROMCLK                      ;EXECUTE
022416  012777  030261  156754      MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022424  104415          ROMCLK                      ;XFR RAM OUTPUT TO "A" REG
022426  012777  010037  156744      MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022434  104415          ROMCLK                      ;F=A
022436  012777  005400  156734      MOV     #BIT11+BIT9+BIT8,@DVSFR ;LOAD BRANCH POINT
022444  017704  156720              MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022450  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022454  012705  000002              MOV     #BIT1,R5        ;SET EXPECTED
022460  020504                      CMP     R5,R4           ;BR POINTS CORRECT?
022462  001401                      BEQ     73$             ;BR IF YES
022464  104006                      HLT     6               ;BRANCH POINTS WRONG
022466  104412          73$:        MSTCLR
022470  012777  000010  156664      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022476  012777  005400  156674      MOV     #BIT11+BIT9+BIT8,@DVSFR ;RESET DV11
022504  017704  156660              MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
022510  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022514  012705  000003              MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
022520  020504                      CMP     R5,R4           ;SET EXPECTED
022522  001401                      BEQ     73$             ;BR POINT OK?
022524  104006                      HLT     6               ;BR IF YES
022526                  73$:                                ;BR POINTS WRONG
                        ;*BRANCH "A" TEST OF ALU 02
                        ;*
022530  104412          MSTCLR                      ;RESET DV11
022532  012777  000010  156624      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
022536  012777  000004  156632      MOV     #BIT2,@DVSRA    ;LOAD DATA
022544  012777  020000  156626      MOV     #RAM,@DVSFR     ;DO A "RAM READ"
022552  104415          ROMCLK                      ;EXECUTE
022554  012777  030261  156616      MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022562  104415          ROMCLK                      ;XFR RAM OUTPUT TO "A" REG
022564  012777  010037  156606      MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022572  104415          ROMCLK                      ;F=A
022574  012777  006000  156576      MOV     #BIT11+BIT10,@DVSFR     ;LOAD BRANCH POINT
022602  017704  156562              MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022606  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022612  012705  000002              MOV     #BIT1,R5        ;SET EXPECTED
022616  020504                      CMP     R5,R4           ;BR POINTS CORRECT?
022620  001401                      BEQ     74$             ;BR IF YES
022622  104006                      HLT     6               ;BRANCH POINTS WRONG
022624  104412          74$:        MSTCLR
022626  012777  000010  156526      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022634  012777  006000  156536      MOV     #BIT11+BIT10,@DVSFR     ;RESET DV11
022642  017704  156522              MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
022646  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022652  012705  000003              MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
022656  020504                      CMP     R5,R4           ;SET EXPECTED
022660  001401                      BEQ     75$             ;BR POINT OK?
022662  104006                      HLT     6               ;BR IF YES
```

```
                                                75$:                    ;BR POINTS WRONG
                                                                        :*BRANCH "9" TEST OF ALU 03
                                                                        :*
 022664  104412                                                         MSTCLR                  ;RESET DV11
 022666  012777  000010  156466                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 022674  012777  000010  156474                                        MOV     #BIT3,@DVSRA    ;LOAD DATA
 022702  012777  020000  156470                                        MOV     #RAM,@DVSFR     ;DO A "RAM READ"
 022710  104415                                                         ROMCLK                  ;EXECUTE
 022712  012777  030261  156460                                        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
 022720  104415                                                         ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
 022722  012777  010037  156450                                        MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
 022730  104415                                                         ROMCLK                  ;F=A
 022732  012777  006400  156440                                        MOV     #BIT11+BIT10+BIT9,@DVSFR        ;LOAD BRANCH POINT
 022740  017704  156424                                                MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
 022744  042704  177774                                                BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 022750  012705  000002                                                MOV     #BIT1,R5        ;SET EXPECTED
 022754  020504                                                        CMP     R5,R4           ;BR POINTS CORRECT?
 022756  001401                                                        BEQ     76$             ;BR IF YES
 022760  104006                                                        HLT     6               ;BRANCH POINTS WRONG
 022762  104412                                75$:    MSTCLR
 022764  012777  000010  156370                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 022772  012777  006400  156400                                        MOV     #BIT11+BIT10+BIT9,@DVSFR        ;RESET DV11
 023000  017704  156364                                                MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
 023004  042704  177774                                                BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
 023010  012705  000003                                                MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
 023014  020504                                                        CMP     R5,R4           ;SET EXPECTED
 023016  001401                                                        BEQ     77$             ;BR POINT OK?
 023020  104006                                                        HLT     6               ;BR IF YES
 023022                                  77$:                          ;BR POINTS WRONG
                                                                        :*BRANCH "A" TEST OF ALU 04
                                                                        :*
 023022  104412                                                         MSTCLR                  ;RESET DV11
 023024  012777  000010  156330                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 023032  012777  000020  156336                                        MOV     #BIT4,@DVSRA    ;LOAD DATA
 023040  012777  020000  156332                                        MOV     #RAM,@DVSFR     ;DO A "RAM READ"
 023046  104415                                                         ROMCLK                  ;EXECUTE
 023050  012777  030261  156322                                        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
 023056  104415                                                         ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
 023060  012777  010037  156312                                        MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
 023066  104415                                                         ROMCLK                  ;F=A
 023070  012777  007000  156302                                        MOV     #BIT11+BIT10+BIT9,@DVSFR        ;LOAD BRANCH POINT
 023076  017704  156256                                                MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
 023102  042704  177774                                                BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 023106  012705  000002                                                MOV     #BIT1,R5        ;SET EXPECTED
 023112  020504                                                        CMP     R5,R4           ;BR POINTS CORRECT?
 023114  001401                                                        BEQ     78$             ;BR IF YES
 023116  104006                                                        HLT     6               ;BRANCH POINTS WRONG
 023120  104412                                78$:    MSTCLR
 023122  012777  000010  156232                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 023130  012777  007000  156242                                        MOV     #BIT11+BIT10+BIT9,@DVSFR        ;RESET DV11
 023136  017704  156226                                                MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
 023142  042704  177774                                                BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
 023146  012705  000003                                                MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
 023152  020504                                                        CMP     R5,R4           ;SET EXPECTED
 023154  001401                                                        BEQ     79$             ;BR POINT OK?
 023156  104006                                                        HLT     6               ;BR IF YES
```

```
                                793$:                           ;BR POINTS WRONG
023160  104400                          SCOPE


                                        ;************************ TEST 101 ******************************
                                        ;*TEST OF BRANCH "B" "RAM OUTPUT 0-14=0".
                                        ;*TEST TO A RAM READ AND "FLOAT" A "1" FROM
                                        ;*RAM 0 TO 14 ; EXPECTING "RAM 014=0" TO BE FALSE.
                                        ;*THEN THE "1" IS SHIFTED INTO BIT15 AND
                                        ;*"RAM 0-14=0" SHOULD BE FALSE.
                                        ;*THIS ALSO TEST "BRB" (RAM OUTPUT BIT15) TRUE.
                                        ;;*************************************************************

                                ;  TEST 101
                                ;--------------
023162  012737  000101  001226  TST101: MOV     #101,TSTNO
023170  012737  023362  001216         MOV     #TST102,NEXT
023176  104412                        MSTCLR                  ;RESET DV11
023200  012703  000001                 MOV     #1,R3           ;SET DATA
023204  012702  076000                 MOV     #BRB+BIT11+BIT10,R2 ;BRB "RAM OUTPUT 0-14=0"
023210  012777  000010  156144         MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
023216  010277  156156                 MOV     R2,@DVSFR       ;LOAD BRB TEST
023222  017704  156142                 MOV     @DVLCR,R4       ;READ TEST POINTS
023226  012705  000001                 MOV     #BIT0,R5        ;SET EXPECTED
023232  042704  177774                 BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
023236  020504                        CMP     R5,R4           ;TRUE?
023240  001401                        BEQ     .+4             ;BR IF YES
023242  104006                        HLT     6               ;RAM OUTPUT 0-14 NOT TRUE AFTER INIT
023244  012705  000003                 MOV     #BIT1+BIT0,R5   ;SET EXPECTED
023250  010377  156122         1$:     MOV     R3,@DVSRA       ;LOAD DATA
023254  012777  02C000  156116         MOV     #RAM,@DVSFR     ;ISSUE RAM READ INSTR
023262  104415                        ROMCLK                  ;
023264  010277  156110                 MOV     R2,@DVSFR       ;BRB TEST
023270  017704  156074                 MOV     @DVLCR,R4       ;READ BR TEST POINTS
023274  042704  177774                 BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
023300  005703                        TST     R3              ;IS 15=1 IN DATA?
023302  100002                        BPL     2$              ;BR IF NO
023304  042705  000002                 BIC     #BIT1,R5        ;IF 15=1 - 0-14=TRUE
023310  020504                 2$:     CMP     R5,R4           ;BRB TEST POINT OK?
023312  001401                        BEQ     3$              ;BR IF YES
023314  104006                        HLT     6               ;BAD TEST POINT
023316  000241                 3$:     CLC                     ;CLEAR CARRY
023320  006103                        ROL     R3              ;MOV BIT TO NEXT RAM POSITION
023322  001352                        BNE     1$              ;HAVE ALL BITS BEEN TESTED?
023324  012705  000001                 MOV     #BIT0,R5        ;SET EXPECTED RESULTS
023330  012777  075400  156042         MOV     #BRB+BIT11+BIT9+BIT8,@DVSFR
023336  017704  156026                 MOV     @DVLCR,R4       ;BRB "RAM OUTPUT 15H"
023342  042704  177774                 BIC     #^C<BIT1+BIT0>,R4
023346  020504                        CMP     R5,R4           ;BRANCH RESULTS OK?
023350  001403                        BEQ     4$              ;BR IF YES
023352  017702  156022                 MOV     @DVSFR,R2       ;SAVE DVSFR FOR TYPEOUT
023356  104006                        HLT     6               ;"RAM OUTPUT 15H" S/B TRUE
023360  104400                 4$:     SCOPE                   ;SCOPE TESTS


                                        ;************************ TEST 102 ******************************
```

```
                                        ;*TEST OF THE RAM WRITE OPERATION.
                                        ;*WRITE ALL SECONDARY REGISTERS FOR ALL LINES
                                        ;*WITH DIFFERENT DATA BY USING THE ROM
                                        ;*AND VERIFY THE DATA BY THE UNIBUS.
                                        ;;**********************************************************

                                ;  TEST 102
                                ;--------------
023362  012737  000102  001226  TST102: MOV     #102,TSTNO
023370  012737  023776  001216          MOV     #TST103,NEXT
023376  104412                        MSTCLR                  ;CLEAR ALL DV11 REGISTERS
023400  012777  000010  155754        MOV     #BIT3,@DVSCR    ;SET SORCE SEC
023406  013700  001400                MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
023412  012702  010077                MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,R2
                                                                ;FUNCTION "F=A+1
023416  012703  020760                MOV     #RAM+BIT8+BIT7+BIT6+BIT5+BIT4,R3
                                                                ;RAM WRITE FROM ALU RESULT.
023422  012704  030361                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,R4
                                                                ;MOVE ALU RESULT TO "A" REG.
023426  005005                        CLR     R5              ;CLEAR LINE NUMBER COUNTER
023430  005001                1$:     CLR     R1              ;ZERO SEC REG POINTER
023432  110577  155734        2$:     MOVB    R5,@DVSRS       ;LOAD LINE
023436  110177  155732                MOVB    R1,@DVSRSH      ;LOAD SEC REG.
023442  012777  177777  155726        MOV     #-1,@DVSRA      ;SET "FOOT PRINT"
023450  042703  000017                BIC     #17,R3          ;CLEAR LINER
023454  050103                        BIS     R1,R3           ;SET LINE
023456  010310                        MOV     R3,(R0)         ;DO "RAM WRITE"
023460  104415                        ROMCLK                  ;
023462  012777  077000  155710        MOV     #BRB+BIT11+BIT10+BIT9,@DVSFR
023470  010546                        MOV     R5,-(SP)        ;SAVE R5
023472  010446                        MOV     R4,-(SP)        ;SAVE R4
023474  010246                        MOV     R2,-(SP)        ;SAVE R2
023476  012705  000001                MOV     #BIT0,R5        ;EXPECTED
023502  017704  155662                MOV     @DVLCR,R4       ;READ BR. RESULT
023506  042704  177774                BIC     #^C<BIT1+BIT0>,R4 ;STRIP JUNK
023512  020504                        CMP     R5,R4           ;WRITE INHIBIT TRUE?
023514  001401                        BEQ     .+4             ;
023516  104006                        HLT     6               ;WRITE INHIBIT FAILED
023520  022777  177777  155650        CMP     #-1,@DVSRA      ;WAS WRITE
023526  001401                        BEQ     .+4             ;REALLY
023530  104000                        HLT     0               ;INHIBTED?
023532  012602                        MOV     (SP)+,R2        ;RESTORE
023534  012604                        MOV     (SP)+,R4        ;REGISTERS
023536  012605                        MOV     (SP)+,R5        ;
023540  012710  020000                MOV     #RAM,(R0)       ;PLACE RAM INSTR IN SFR
023544  050110                        BIS     R1,(R0)         ;PLACE SEC REG POINTER IN SFR
023546  104415                        ROMCLK                  ;EXECUTE INSTR.
023550  010210                        MOV     R2,(R0)         ;"F=A+1"
023552  104415                        ROMCLK                  ;EXECUTE
023554  042703  000017                BIC     #17,R3          ;CLEAR SEC REG POINTER
023560  050103                        BIS     R1,R3           ;SET SEC REG POINTER
023562  010310                        MOV     R3,(R0)         ;RAM WRITE FROM ALU RESULT
023564  104415                        ROMCLK                  ;EXECUTE
023566  012777  077000  155604        MOV     #BRB+BIT11+BIT10+BIT9,@DVSFR
023574  010546                        MOV     R5,-(SP)        ;TEST WRITE
023576  010446                        MOV     R4,-(SP)        ;INHIBIT
```

```
                                                        R2,-(SP)        ;FALSE!
                                                        (SP)+,SFR_R2
                                                        #BIT1+BIT0,R5   ;EXPECTED
                                                        R5,R4           ;RES. RESULTS
                                                        #BIT1+BIT0,R4   ;RIP COM.
                                                        R5,R4           ;GOOD?
                                                        10$
                                                        #6,R0           ;WRITE INHIBIT S B FALSE
                                                        (SP)+,R2        ;RESTORE
                                                        (SP)+,R4        ;REGISTERS
                                                        (SP)+,R5
                                                        R4,@(R0)+       ;MOVE ALU RESULT TO A REG  UPDATED DATA
                                                        #CLA            ;EXECUTE
                                                        R1              ;UPDATE SEC REG POINTER
                                                        #16.,R1         ;ALL REGISTERS DONE?
                                                        20$             ;BR IF NO
                                                        #MSCAN,-(SP)    ;SAVE RO
                                                        #PERFORM,SET$CAN ;UPDATE MSCAN
                                                        20$             ;BNE NO, UPDATE
                                                        (SP)+,R0        ;RESTORE RO
                                                        #CLA            ;FOR MSCAN TO RAM 0-3 0-15
                                                        R5              ;UPDATE LINE NUMBER COUNTER.
                                                        #16.,R5         ;ALL LINES DONE?
                                                        30$             ;BR IF NO
                                                        #6,R0           ;ZERO SEC REG POINTER
                                                        #1,R5           ;UPDATE LINE NUMBER POINTER
                                                        #MSCAN,R4       ;SRC POINTER FOR MSCAN
                                                        #3,R3           ;HIGH BYTE POINTER  USE REG SEL
                                                        #1,R2           ;SRA POINTER  ACCESS RO
                                                        #0,R1           ;EXPECTED DATA
                                                        #0,R0           ;LOAD SEC REG SEL
                                                        #CLA            ;LOAD LINE NO.
                                                        R4              ;LOAD RAM RESULT
                                                        R1              ;WAS RAM "WRITTEN" CORRECTLY?
                                                        40$             ;BR IF RAM DATA OK
                                                        #BIT0,R1        ;RAM WAS "WRITTEN" INCORRECTLY
                                                        R0              ;UPDATE SEC REG POINTER AND DATA EXPECTED
                                                        #16.,R0         ;ALL SEC REGISTERS DONE?
                                                        40$             ;BR IF NO
                                                        #0,R0           ;ZERO SEC REG POINTER
                                                        R5              ;UPDATE LINE NUMBER POINTER
                                                        #16.,R5         ;ALL LINES DONE?
                                                        50$             ;BR IF NO
                                                        #1,R5           ;SCOPE TEST
SCOPE
```

```
;********************** TEST 103 ******************************
;*BASIC TEST FOR THE "BIS" OPERATION"
;*POLYNOMIAL SELECTION TABLE:
;*FROM OUTPUT BIT04 BIT03        POLY
;*                0     0        LRC 8
;*                0     1        CRC 16
;*                1     1        CRC CCITT
;*************************************************************

;********************** TEST 103 ******************************
```

```
                                        ;*TEST OF LRS 8.
                                        ;*PATTERNS ARE:
                                        ;*    DATA     B REG    BCC   EXPECTED
                                        ;*     0        0        0       0
                                        ;*     1        1        1       0
                                        ;*********************************************************

                                ;  TEST 103
                                ;-------------
                        ST103:  MOV     #103,TSTNO
                                MTS     #104,NEXT
                        MSTCLR  MOV     #CLR,SCR        ;RESET DVII
                                MOV     #BIT3,SCR       ;SET SOURCE SEL
                                MOV     #0,R0           ;INSERT CVSFR POINTER
                                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,R2
                                MOV     #BCC,-(R0)      ;DO A "BCC" INSTRUCTION USING "-(R0)"
                                MOV     R0,-(R0)        ;EXECUTE
                                MOV     R2,-(R0)        ;XFR BCC REG TO DVRIC
                                MOV     R0,-(R0)        ;EXECUTE
                                CLR     R5              ;SET EXPECTED TO 0
                                MOV     DVRIC,R4        ;READ RESULTS OF BCC REG.
                                BEQ     1$              ;BR IF =0
                                HLT     6               ;BCC REG NOT =0
                        1$:     MOV     #ALU+BIT4+BIT3+BIT2,-(R0)
                                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,-(R0)
                                MOV     R0,-(R0)        ;XFR ALU RESULT TO B REGISTER
                                MOV     #BCC,-(R0)      ;DO A "BCC" OPR
                                MOV     R0,-(R0)        ;EXECUTE
                                MOV     R2,-(R0)        ;XFR BCC REG TO DVRIC
                                MOV     R0,-(R0)        ;EXECUTE
                                MOV     #377,R5         ;IF EXPECTED RESULTS SHALL BE LOW BYTE
                                MOV     DVRIC,R4        ;READ RESULTS
                                BIC     #400,R4         ;MSK BCC OPR WORK
                                BEQ     2$              ;BR IF OK
                                HLT     6               ;EITHER "BCC" OPR FAILED OR BITS  INCORRECT IN LOW BYTE
                        2$:     MOV     #CLR,SCR        ;RESET INTERNAL REG (BCC,ALU)
                                MOV     #BIT3,SCR       ;SET SOURCE SEL.
                                MOV     #ALU+BIT4+BIT3+BIT2,-(R0)
                                MOV     R0,-(R0)        ;ALU "F=-1"
                                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,-(R0)
                                MOV     R0,-(R0)        ;XFR ALU RESULT TO "A" REGISTER
                                MOV     #BCC,-(R0)      ;BCC OPERATION
                                MOV     R0,-(R0)        ;EXECUTE
                                MOV     R2,-(R0)        ;XFR BCC TO DVRIC
                                MOV     R0,-(R0)        ;EXECUTE
                                MOV     DVRIC,R4        ;READ RESULTS
                                CMP     R5,R4           ;BCC REG = B =377
                                BEQ     3$              ;BR IF OK
                                HLT     6               ;BCC REG NOT =377
                                                        ;READ OF DATA FROM "A" REG LED FAILED
                        3$:     MSTCLR                  ;CLEAR INTERNAL REGISTERS
                                MOV     #BIT3,SCR       ;SET SOURCE SEL
                                MOV     #ALU+BIT4+BIT3+B[illegible],-(R0)
```

```
                                                        ;ALU "F=-1"
030381                                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
                                        ROMCLK          ;XFR ALU RESULT TO "A" REG
030381                                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
                                        ROMCLK          ;XFR ALU RESULT TO "B" REG
000000                                  MOV     #BCC,(R0)       ;SO BCC OPR
                                        ROMCLK          ;
                                        MOV     R2,(R0)         ;XFR BCC TO D.BIC
                                        ROMCLK          ;ROMCLK
                                        CLR     R0              ;EXPECT 0'S
100100                                  MOV     #4$,BIC,R4      ;BCC RESULT
                                        BEQ     4$              ;BR IF =0
                                        HALT    5               ;"LRL" XOR TEST FAILED
                                8$:     SCOPE                   ;SCOPE TEST

                                        ;********************** TEST 104 ****************************
                                        ;*TEST IF BCC(NORMAL "CRC 16"
                                        ;*TEST THAT BIT 5 OF THE "B" REG APPEAR
                                        ;*IN BITS 1-5 OF THE BCC REG.
                                        ;************************************************************

                                ;  TEST 104
                                ;-------------
                                $TST4:  MOV     #104,TESTNO
                                        #$TST4,SCOPE
                                        ROMCLK          ;RESET D.BIC
                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT3,R0        ;SET UP SCOPE SE
                                        MOV     #0,@#177572     ;SET DVSFR COUNTER
                                        MOV     #252525,R0      ;LOAD "A"
                                        MOV     R0,@#RAM        ;RAM REAC
                                        ROMCLK          ;EXECUTE
030261                                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,R0
                                        ROMCLK          ;XFR RAM OUTPUT TO "A" REG.
030262                                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,R0
                                        ROMCLK          ;XFR RAM OUTPUT TO "B" REG.
000003                  155036          MOV     #BIT3,@#SRA     ;SELECT POLYNOMIAL "CRC16"
000000                                  MOV     #RAM,(R0)       ;READ DATA
                                        ROMCLK          ;
                                        MOV     #BIT3,R5        ;SET EXPECTED RESULTS
000000                          8$:     MOV     #BCC,R0         ;LOAD OPR "CRC16"
                                        ROMCLK          ;EXECUTE
030346                                  MOV     #XFR+BIT7+BIT5+BIT2+BIT1,R0
                                        ROMCLK          ;XFR BCC REG TO D.BIC
                                        MOV     @#BIC,R4        ;READ RESULTS
000010                                  MOV     #8.,R3          ;PREPARE
                                4$:     ROR     R4              ;TO
                                        ROR     R4              ;POSITION
                                        DECR    R3              ;RESULT
                                        BNE     3$              ;OF BCC OPERATION
                                        CMP     R5,R4           ;CID CRC16 WORK
                                        BEQ     4$              ;BR IF YES
                                        HALT    6               ;INCORRECT BCC RESULTS
010026                          9$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
                                        ROMCLK          ;ALU "F=A+B"
030362                                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,R0
                                        ROMCLK          ;XFR ALU RESULT TO "B" REG
```

E08


```
                                                        ;UPDATE DATA COMPARE
                                                        ;ALL PATH DONE
                                                        ;BR IF NO
                                                        ;SCOPE TEST


                                                        ;+++++++++++++++++++++++ TEST 105 +++++++++++++++++++++++++++++
                                                        ;+TEST OF THE BCC OPERATION USING
                                                        ;+VARIOUS CRC'S FOR THE POLYNOMIAL
                                                        ;+INCLUDING PATTERNS PRESTORED IN
                                                        ;+SILO'S AS WORDS AS POSSIBLE
                                                        ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                    ;  TEST 105
                                    ;--------------
                                    START:  CLR     R0              ;GET POLYNOMIA
                                            MOV     #177570,R1      ;GET REGISTER POINTER
                                            MOV     #BIT15!BIT14!BIT13!BIT12!BIT11!BIT10!BIT9!BIT8!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2!BIT1!BIT0,(R1)
                                                                    ;SET FOR XFER FROM BCC TO D.REG
                                            MOV     #10.,R3         ;IF SW09=1
                                    1$:     MOV     #1,(R1)         ;LOAD A MSTCLR
                                            MOV     #BIT3!BIT0,(R1) ;SET SOURCE SEL
                                            MOV     (R1),R5         ;READ CALCULATED BCC (SOFTWARE
                                            MOV     R5,SIMBCC       ;BCC AND CALCULATE A NEW BCC
                                                                    ;SHIFTS PERFORMED
                                                                    ;DATA CHAR
                                                                    ;PREVIOUS BCC
                                            MOV     R3,L.DATA       ;GO AND LOAD DATA INTO THE D.R
                                                                    ;DATA TO THE A REGISTER
                                                                    ;DATA TO THE B REGISTER
                                                                    ;POLY SELT. (RAMCLK=0)

                                            MOV     #BCC,R0
                                            MOV     #1,(R1)         ;LOAD A MSTCLR
                                            MOV     R2,(R0)         ;GO A BCC CALCULATION (RAMCLK=1)
                                                                    ;GO A DATAXFR FROM BCC REG TO D.REG
                                                                    ;
                                            CMP     CALBCC,R5       ;PUT SOFTWARE BCC INTO R5 REGISTER
                                            MOV     #BCC,R0         ;PUT FROM BCC INTO REGISTER (RAMCLK=0)
                                            CMP     #5,R4           ;DOES EXPECTED = FOUND
                                            BEQ     6$              ;BR IF OK!
                                            ILT     5               ;BCC CALCULATION ERROR
                                    6$:     MOV     #1,(R1)         ;LOAD A MSTCLR
                                            MOV     #BIT3,(R1)      ;SET IF SW09=1
                                    2$:     MOV     #1,(R1)         ;ISSUE A MSTCLR
                                            MOV     #BIT3,(R1)      ;SET SOURCE SEL
                                            MOV     CALBCC          ;READ CALCULATED BCC (SOFTWARE
                                            MOV     R5,SIMBCC       ;BCC AND CALCULATE A NEW BCC
                                                                    ;SHIFTS PERFORMED
                                                                    ;DATA CHAR
                                                                    ;PREVIOUS BCC
                                            MOV     R3,L.DATA       ;GO AND LOAD DATA INTO THE D.R
                                                                    ;DATA TO THE A REGISTER
                                                                    ;DATA TO THE B REGISTER
                                                                    ;POLY SELT. (RAMCLK=0)
```

F08


```
                    000000                          MOV     #BCC,@RC)       ;GO A BCC CALCULATION (HROW)
                                                    ROMCLK
                                                    MOV     R2,@RC)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                    ROMCLK                  ;
                    031666                          MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                    154046                          MOV     @DVRIC,R4       ;PUT HROW BCC INTO RECEIVED
                                                    CMP     R5,R4           ;DOES EXPECTED = FOUND ?
                                                    BEQ     66$             ;BR IF OK!
                                                    HLT     6               ;BCC CALCULATION ERROR
                                            66$:    SCOPI                   ;SW09=1?
                    025004  001220                  MOV     #4$,LOOP        ;SET IF SW09=1
                                             4$:    MSTCLR                  ;ISSUE A MSTCLR
                    060010  154046                  MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                    CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                    031474                          JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                                            ;SHIFTS PERFORMED
                                                    ;B'11111111'                    ;DATA CHAR
                                                    0                       ;PREVIOUS BCC
                    031420                          JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE B/A
                                                    ;B'11111111'                    ;DATA TO THE A REGISTER
                                                    0                       ;DATA TO THE B REGISTER
                                                    10                      ;POLY SELT. (RAMOUTPUT)
                    060000                          MOV     #BCC,(R0)
                                                    ROMCLK
                                                    MOV     R2,(R0)         ;DO A BCC CALCULATION (HROW)
                                                    ROMCLK                  ;DO A DATAXFR FROM BCC REG TO DVRIC
                    031666                          MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                    154046                          MOV     @DVRIC,R4       ;PUT HROW BCC INTO RECEIVED
                                                    CMP     R5,R4           ;DOES EXPECTED = FOUND ?
                                                    BEQ     67$             ;BR IF OK!
                                                    HLT     6               ;BCC CALCULATION ERROR
                                            67$:    SCOPI                   ;SW09=1?
                    025104  001220                  MOV     #5$,LOOP        ;SET IF SW09=1
                                             5$:    MSTCLR                  ;ISSUE A MSTCLR
```

```
                                                MOV     #BIT3,2DVSCR    ;SET SOURCE SEL
                                                CLR     R4              ;ZERO CALCULATED BCC SOFTWARE
                                                SR      R5,S1M3CC       ;GO AND CALCULATE A NEW BCC
                                                                        ;SHIFTS PERFORMED
                                                B 00000001:             ;DATA CHAR
                                                                        ;PREVIOUS BCC
                                                MOVB    R3,L.DATA       ;GO AND LOAD DATA INTO THE DMA
                                                B 00000001:             ;DATA TO THE A REGISTER
                                                                        ;DATA TO THE B REGISTER
                                                                        ;POLY SELT. (RAMOUTPUT)
                                                MOV     #3CC,.P2
                                                MOVCLK
                                                MOV     R2,.BCC         ;GO A BCC CALCULATION (HARD)
                                                MOVCLK                  ;GO A DATAXFR FROM BCC REG TO D.BCC
                                                                        ;
                                                CMP     SAL800,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     #1,R4           ;PUT 1 INTO BCC INTO REGISTER
                                                BNE     6$              ;BCCES EXPECTED = FOUND
                                                INC     R4              ;GO IF OK!
                                                6$:                     ;BCC CALCULATION ERROR
                                                                        ;SWC9=1
                                                BIT     #S1,L.CCR       ;TEST IF SWC9=1
                                                BEQ     8$              ;ISSUE A MSTCLR
                                                MOV     #BIT3,2DVSCR    ;SET SOURCE SEL
                                                CLR     R4              ;ZERO CALCULATED BCC SOFTWARE
                                                SR      R5,S1M3CC       ;GO AND CALCULATE A NEW BCC
                                                                        ;SHIFTS PERFORMED
                                                B 00010000:             ;DATA CHAR
                                                                        ;PREVIOUS BCC
                                                MOVB    R3,L.DATA       ;GO AND LOAD DATA INTO THE DMA
                                                B 00010000:             ;DATA TO THE A REGISTER
                                                                        ;DATA TO THE B REGISTER
                                                                        ;POLY SELT. (RAMOUTPUT)
                                                MOV     #3CC,.P2
                                                MOVCLK                  ;GO A BCC CALCULATION (HARD)
                                                MOV     R2,.BCC         ;GO A DATAXFR FROM BCC REG TO D.BCC
                                                MOVCLK                  ;
                                                CMP     SAL800,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     #1,R4           ;PUT 1 INTO BCC INTO REGISTER
                                                BNE     6$              ;BCCES EXPECTED = FOUND
                                                INC     R4              ;GO IF OK!
                                                6$:                     ;BCC CALCULATION ERROR
                                                                        ;SWC9=1
                                                BIT     #S1,L.CCR       ;TEST IF SWC9=1
                                                BEQ     8$              ;ISSUE A MSTCLR
                                                MOV     #BIT3,2DVSCR    ;SET SOURCE SEL
                                                CLR     R4              ;ZERO CALCULATED BCC SOFTWARE
                                                SR      R5,S1M3CC       ;GO AND CALCULATE A NEW BCC
                                                                        ;SHIFTS PERFORMED
                                                B 00010000:             ;DATA CHAR
                                                                        ;PREVIOUS BCC
                                                MOVB    R3,L.DATA       ;GO AND LOAD DATA INTO THE DMA
                                                B 00010000:             ;DATA TO THE A REGISTER
                                                                        ;DATA TO THE B REGISTER
                                                                        ;POLY SELT. (RAMOUTPUT)
                                                MOV     #3CC,.P2
                                                MOVCLK                  ;GO A BCC CALCULATION (HARD)
```

```
                                                MOV     R2,.RC)         ;DO A DATAXFR FROM BCC REG TO D.BUS
                                                RCMCLK
                                                MOV     CAL.BCC,R5      ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @RBCC,R4        ;PUT HRDW BCC INTO REGISTERED
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND
                                                BEQ     .+4             ;BR IF OK!
                                                HLT     5               ;BCC CALCULATION ERROR
                                                SCOP1                   ;SW09=1?
                                                MOV     #SS$,.LOOK      ;SET IF SW09=1
                                                MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,SDVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                SR      R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                .                       ;SHIFTS PERFORMED
                                                .=.+1000000             ;DATA CHAR
                                                                        ;PREVIOUS BCC
                                                SR      R3,#.DATA       ;GO AND LOAD DATA INTO THE D/I/
                                                .=.+1000004             ;DATA TO THE A REGISTER
                                                                        ;DATA TO THE B REGISTER
                                                                        ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,.RC)
                                                RCMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,.RC)         ;DO A DATAXFR FROM BCC REG TO D.BUS
                                                RCMCLK
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @RBCC,R4        ;PUT HRDW BCC INTO REGISTERED
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND
                                                BEQ     .+4             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                                SCOP1                   ;SW09=1?
                                                MOV     #SS$,.LOOK      ;SET IF SW09=1
                                                MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,SDVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                SR      R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                .                       ;SHIFTS PERFORMED
                                                .=.+1000000             ;DATA CHAR
                                                                        ;PREVIOUS BCC
                                                SR      R3,#.DATA       ;GO AND LOAD DATA INTO THE D/I/
                                                .=.+1000004             ;DATA TO THE A REGISTER
                                                                        ;DATA TO THE B REGISTER
                                                                        ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,.RC)
                                                RCMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,.RC)         ;DO A DATAXFR FROM BCC REG TO D.BUS
                                                RCMCLK
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @RBCC,R4        ;PUT HRDW BCC INTO REGISTERED
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND
                                                BEQ     72$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                                SCOP1                   ;SW09=1?
                                                MOV     #100$,.LOOK     ;SET IF SW09=1
                                                MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,SDVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
```

```
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                                        ;SHIFTS PERFORMED
                                                .BYTE   ^B01010000      ;DATA CHAR
                                                .BYTE   0               ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                .BYTE   ^B01010000      ;DATA TO THE A REGISTER
                                                .BYTE   0               ;DATA TO THE B REGISTER
                                                .BYTE   10              ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,-(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,-(R0)        ;DO A DATAXFR FROM BCC REG TO D.BUS
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @#177570,R4     ;PUT HRDW BCC INTO RECEIVERC
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     73$             ;BR IF OK!
                                                HALT    6               ;BCC CALCULATION ERROR
                                        73$:    SCOPI                   ;SW09=1?
                                                MOV     #101$,LOCK      ;SET IF SW09=1
                                        101$:   MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                .BYTE   ^B11010000      ;SHIFTS PERFORMED
                                                                        ;DATA CHAR
                                                .BYTE   0               ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                .BYTE   ^B11010000      ;DATA TO THE A REGISTER
                                                .BYTE   0               ;DATA TO THE B REGISTER
                                                .BYTE   10              ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,-(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,-(R0)        ;DO A DATAXFR FROM BCC REG TO D.BUS
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @#177570,R4     ;PUT HRDW BCC INTO RECEIVERC
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     74$             ;BR IF OK!
                                                HALT    6               ;BCC CALCULATION ERROR
                                        74$:    SCOPI                   ;SW09=1?
                                                MOV     #102$,LOCK      ;SET IF SW09=1
                                        102$:   MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                .BYTE   ^B11000000      ;SHIFTS PERFORMED
                                                                        ;DATA CHAR
                                                .BYTE   0               ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                .BYTE   ^B11000000      ;DATA TO THE A REGISTER
                                                .BYTE   0               ;DATA TO THE B REGISTER
                                                .BYTE   10              ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,-(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,-(R0)        ;DO A DATAXFR FROM BCC REG TO D.BUS
                                                ROMCLK                  ;
```

```
...A-S MACY11 30.732  17-SEP-76  14:02  PAGE 10
...DZ.P11     DV11 DEVICE DIAGNOSTICS.   COPYRIGHT 1976  DIGITAL EQUIP. CORP.

                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND?
                                                BEQ     7$              ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                    7$:         SCOPE                   ;SWR9=1?
                                                SCOPE


                                                ;********************** TEST 106 ****************************
                                                ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
                                                ;*THROUGH THE BCC GENERATION LOGIC.
                                                ;*THE POLYNOMIAL USED WILL BE LRC8.
                                                ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                                ;*EACH CHARACTER ->NOT ZEROED OUT*-
                                                ;;************************************************************

                                                ;  TEST 106
                                                ;--------------
026100  012737  000106  001226  TST106: MOV     #106,TSTNO
026106  012737  026256  001216          MOV     #TST10P,NEXT
026114  012737  000200  001646          MOV     #LRC8,XPOLY     ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
026122  012702  000345                  MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT1,R2
026126  010037  001400                  MOV     DVSFR,@0        ;LOAD DATA XFER FROM BCC TO DVRIC
026132  005001                          CLR     R1              ;SET DATA POINTER TO 0
026134  005037  001652                  CLR     CALBCC          ;ZERO SOFTWARE BCC
026140  104412                  1$:     MSTCLR                  ;CLEAR THE DV11
026142  012777  000010  153212          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
026150  010137  026202                  MOV     R1,2$           ;LOAD SOFTWARE CHAR
026154  010137  026212                  MOV     R1,4$           ;LOAD HRDW CHAR
026160  013737  001652  026204          MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
026166  013737  001652  026214          MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
026174  004537  001474                  JSR     R5,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
026200  000010                          8.                      ;EIGHT SHIFTS
026202  000001                  2$:     .BLKW 1                 ;DATA
026204  000001                  3$:     .BLKW 1                 ;PREVIOS BCC
026206  004537  001420                  JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
026212  000001                  4$:     .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
026214  000001                  5$:     .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
026216  000000                          0                       ;TO BE LEFT IN THE RAM OUTPUT REG
026220  012710  060000                  MOV     #BCC,(R0)
026224  104415                          ROMCLK                  ;DO A BCC OPERATION
026226  010210                          MOV     R2,(R0)         ;DO A DATA XFER OPR.
026230  104415                          ROMCLK                  ;
026232  013705  001652                  MOV     CALBCC,R5       ;GET GOOD BCC
026236  017704  153124                  MOV     @DVRIC,R4       ;GET ???? BCC
026242  020504                          CMP     R5,R4           ;ARE THEY THE SAME?
026244  001401                          BEQ     6$              ;BR IF YES
026246  104006                          HLT     6               ;BCC ERROR
026250  105201                  6$:     INCB    R1              ;UPDATE DATA CHAR
026252  001332                          BNE     1$              ;BR IF NOT ALL DATA DONE.
026254  104400                          SCOPE                   ;SCOPE THIS TEST


                                                ;********************** TEST 107 ****************************
                                                ;*TEST TO RUN A BINARY COUNT (000-377 PATTERN
```

```
                                        ;*THROUGH THE BCC GENERATION LOGIC.
                                        ;*THE POLYNOMIAL USED WILL BE CRC16 .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;**********************************************************

                                ;  TEST 107
                                ;--------------
026256  012737  000107  001226  TST107: MOV     #107,TSTNO
026264  012737  026434  001216          MOV     #TST110,NEXT
026272  012737  120001  001646          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
026300  012702  000346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
026304  013700  001400                  MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DV11C
026310  005001                          CLR     R1              ;SET DATA POINTER TO 0
026312  005037  001652                  CLR     CALBCC          ;ZERO SOFTWARE BCC
026316  104412                  1$:     MSTCLR                  ;CLEAR THE DV11
026320  012777  000010  153034          MOV     #BIT3,@DVSOR    ;SET SOURCE SEL
026326  010137  026360                  MOV     R1,2$           ;LOAD SOFTWARE CHAR
026332  010137  026370                  MOV     R1,4$           ;LOAD HRDW CHAR
026336  013737  001652  026362          MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
026344  013737  001652  026372          MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
026352  004537  001474                  JSR     R5,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
026356  000010                          8.                      ;EIGHT SHIFTS
026360  000001                  2$:     .BLKW 1                 ;DATA
026362  000001                  3$:     .BLKW 1                 ;PREVIOS BCC
026364  004337  001420                  JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
026370  000001                  4$:     .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
026372  000001                  5$:     .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
026374  000010                          10                      ;TO BE LEFT IN THE RAM OUTPUT REG
026376  012710  060000                  MOV     #BCC,(R0)
026402  104415                          ROMCLK                  ;DO A BCC OPERATION
026404  010210                          MOV     R2,(R0)         ;DO A DATA XFER OPR.
026406  104415                          ROMCLK                  ;
026410  013705  001652                  MOV     CALBCC,R5       ;GET GOOD BCC
026414  017704  152746                  MOV     @DVRIC,R4       ;GET ???? BCC
026420  020504                          CMP     R5,R4           ;ARE THEY THE SAME?
026422  001401                          BEQ     6$              ;BR IF YES
026424  104006                          HLT     6               ;BCC ERROR
026426  105201                  6$:     INCB    R1              ;UPDATE DATA CHAR
026430  001???                          BNE     1$              ;BR IF NOT ALL DATA DONE.
026432  104???                          SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 110 ****************************
                                        ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
                                        ;*THROUGH THE BCC GENERATION LOGIC.
                                        ;*THE POLYNOMIAL USED WILL BE CRC.CCITT .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;**********************************************************

                                ;  TEST 110
                                ;--------------
026434  012737  000110  001226  TST110: MOV     #110,TSTNO
026442  012737  026612  001216          MOV     #TST111,NEXT
026450  012737  120010  001646          MOV     #CRC.CCITT,XPOLY        ;LOAD POLYNOMIAL FOR SOFTWARE CAL
```

L08


```
 5313  026456  012702  030346                 MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
 5314  026462  013700  001400                 MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DV5IC
 5315  026466  005001                         CLR     R1              ;SET DATA POINTER TO 0
 5316  026470  005037  031652                 CLR     CALBCC          ;ZERO SOFTWARE BCC
 5317  026474  104412              1$:        MSTCLR                  ;CLEAR THE DV11
 5318  026476  012737  000010  152656         MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5319  026504  010137  026536                 MOV     R1,2$           ;LOAD SOFTWARE CHAR
 5320  026510  010137  026546                 MOV     R1,4$           ;LOAD HRDW CHAR
 5321  026514  013737  031652  026540         MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
 5322  026522  013737  031652  026550         MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
 5323  026530  004537  031474                 JSR     R5,SIM3CC       ;HAVE SOFTWARE GET THE RIGHT BCC
 5324  026534  000010                         9.                      ;EIGHT SHIFTS
 5325  026536  000001              2$:        .BLKW 1                 ;DATA
 5326  026540  000001              3$:        .BLKW 1                 ;PREVIOS BCC
 5327  026542  004337  031420                 JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
 5328  026546  000001              4$:        .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
 5329  026550  000001              5$:        .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
 5330  026552  000030                         30                      ;TO BE LEFT IN THE RAM OUTPUT REG
 5331  026554  012710  060000                 MOV     #BCC,(R0)
 5332  026560  104415                         ROMCLK                  ;DO A BCC OPERATION
 5333  026562  010210                         MOV     R2,(R0)         ;DO A DATA XFER OPR.
 5334  026564  104415                         ROMCLK                  ;
 5335  026566  013705  031652                 MOV     CALBCC,R5       ;GET GOOD BCC
 5336  026572  017704  152570                 MOV     @DVRIC,R4       ;GET ???? BCC
 5337  026576  020504                         CMP     R5,R4           ;ARE THEY THE SAME?
 5338  026600  001401                         BEQ     6$              ;BR IF YES
 5339  026602  104006                         HLT     6               ;BCC ERROR
 5340  026604  105201              6$:        INCB    R1              ;UPDATE DATA CHAR
 5341  026606  001332                         BNE     1$              ;BR IF NOT ALL DATA DONE.
 5342  026610  104400                         SCOPE                   ;SCOPE THIS TEST
```

```
                                        ;********************** TEST 111 ****************************
                                        ;*TEST THAT SETTING BIT9!BIT7 AND BIT9!BIT6
                                        ;*RECV IE AND RECV INTR PRODUCE AN INTERUPT ON VECTOR "A"
                                        ;;************************************************************

                                ;  TEST 111
                                ;-------------
026612  012737  000111  001226 TST111: MOV     #111,TSTNO
026620  012737  026762  001216         MOV     #TST112,NEXT
026626  012737  000340  177776         MOV     #340,PS          ;LOCK OUT CPU INTERUPTS
026634  104412                         MSTCLR                   ;ISSUE DVRESET
026636  004537  031654                 JSR     R5,SETVEC        ;GO SET VECTOR "A" AND "B"
026642  031676                         NO.ATRAP                 ;"A"
026644  031702                         NO.BTRAP                 ;"B"
026646     340     340                 .BYTE   340,340          ;PRIO. AT 7
026650  005037  177776                 CLR     PS               ;ZERO CPU PRIO.
026654  012777  001200  152500         MOV     #BIT9!BIT7,@DVSCR        ;SET AN INTERUPT RELATIVE BIT.
026662  000240                         NOP                      ;WAIST TIME
026664  012777  001100  152470         MOV     #BIT9!BIT6,@DVSCR        ;SET THE ALTERNATE RELATIVE BIT.
026672  000240                         NOP                      ;WAIST
026674  005077  152462                 CLR     @DVSCR           ;ZERO REG
026700  004537  031654                 JSR     R5,SETVEC        ;GO RESET VECTORS "A" AND "B"
026704  026744                         2$                       ;"A"
026706  026756                         3$                       ;"B"
026710     340     340                 .BYTE   340,340          ;PRIO. AT 7
026712  052777  001300  152442         BIS     #BIT9!BIT7!BIT9!BIT6,@DVSCR
026720  000240                         NOP                      ;SET BOTH INTERUPT RELATIVE BITS.
026722  104007                         HLT     7                ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
026724  004537  031654          1$:    JSR     R5,SETVEC        ;RESET VECTORS
026730  031676                         NO.ATRAP                 ;"A"
026732  031702                         NO.BTRAP                 ;"B"
026734     340     340                 .BYTE   340,340
026736  005077  152420                 CLR     @DVSCR           ;DSABLE DV11
026742  104400                         SCOPE                    ;SCOPE THIS TEST.
026744  012706  001200          2$:    MOV     #STACK,SP        ;RESET STACK
026750  005077  152406                 CLR     @DVSCR           ;DSABLE DV11
026754  000763                         BR      1$               ;RETURN
026756  104012          3$:    HLT     12               ;VECTOR HERE WAS WRONG SIDE
026760  000771                         BR      2$

                                        ;********************** TEST 112 ****************************
                                        ;*TEST THAT SETTING BIT12 AND BIT10
                                        ;*STORE IE AND NPR STAT OVFLOW PRODUCE AN INTERUPT ON VECTOR "B"
                                        ;;************************************************************

                                ;  TEST 112
                                ;-------------
026762  012737  000112  001226 TST112: MOV     #112,TSTNO
026770  012737  027132  001216         MOV     #TST113,NEXT
026776  012737  000340  177776         MOV     #340,PS          ;LOCK OUT CPU INTERUPTS
027004  104412                         MSTCLR                   ;ISSUE DVRESET
027006  004537  031654                 JSR     R5,SETVEC        ;GO SET VECTOR "A" AND "B"
027012  031676                         NO.ATRAP                 ;"A"
027014  031702                         NO.BTRAP                 ;"B"
027016     340     340                 .BYTE   340,340          ;PRIO. AT 7
```

```
                                                        CLR     PS              ;ZERO CPU PRIO.
                                                        MOV     #BIT12,@DVSCR   ;SET AN INTERUPT RELATIVE BIT.
                                                        NOP                     ;WAIST TIME
                                                        MOV     #BIT10,@DVSCR   ;SET THE ALTERNATE RELATIVE BIT.
                                                        NOP                     ;WAIST
                                                        CLR     @DVSCR          ;ZERO REG
                                                        JSR     R5,SETVEC       ;GO RESET VECTORS "A" AND "B"
                                                        3$                      ;"A"
                                                        2$                      ;"B"
                                                        .BYTE   340,340         ;PRIO. AT 7
                                                        BIS     #BIT12!BIT10,@DVSCR
                                                        NOP                     ;SET BOTH INTERUPT RELATIVE BITS.
                                                        HLT     10              ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
                                        1$:             JSR     R5,SETVEC       ;RESET VECTORS
                                                        NO.ATRAP                ;"A"
                                                        NO.BTRAP                ;"B"
                                                        .BYTE   340,340
                                                        CLR     @DVSCR          ;DSABLE DV11
                                                        SCOPE                   ;SCOPE THIS TEST.
                                        2$:             MOV     #STACK,SP       ;RESET STACK
                                                        CLR     @DVSCR          ;DSABLE DV11
                                                        BR      1$              ;RETURN
                                        3$:             HLT     11              ;VECTOR HERE WAS WRONG SIDE
                                                        BR      2$

                                                ;********************** TEST 113 ****************************
                                                ;*TEST THAT SETTING BIT15!BIT9 AND BIT13!BIT9
                                                ;*NPR STAT INTR AND NPR STAT IE PRODUCE AN INTERUPT ON VECTOR "3"
                                                ;*****************************************************************

                                                ;  TEST 113
                                                ;--------------
027132  012737  000113  001226  TST113: MOV     #113,TSTNO
027140  012737  027310  001216          MOV     #TST114,NEXT
027146  012737  000340  177776          MOV     #340,PS         ;LOCK OUT CPU INTERUPTS
027154  104412                        MSTCLR                  ;ISSUE DVRESET
027156  004537  031654                JSR     R5,SETVEC       ;GO SET VECTOR "A" AND "B"
027162  031676                        NO.ATRAP                ;"A"
027164  031702                        NO.BTRAP                ;"B"
027166     340     340                .BYTE   340,340         ;PRIO. AT 7
027170  005037  177776                CLR     PS              ;ZERO CPU PRIO.
027174  012777  101000  152160        MOV     #BIT15!BIT9,@DVSCR      ;SET AN INTERUPT RELATIVE BIT.
027202  000240                        NOP                     ;WAIST TIME
027204  012777  021000  152150        MOV     #BIT13!BIT9,@DVSCR      ;SET THE ALTERNATE RELATIVE BIT.
027212  000240                        NOP                     ;WAIST
027214  005077  152142                CLR     @DVSCR          ;ZERO REG
027220  012777  001000  152134        MOV     #BIT9,@DVSCR    ;SET SYS MAINT ENABLE
027226  004537  031654                JSR     R5,SETVEC       ;GO RESET VECTORS "A" AND "B"
027232  027304                        3$                      ;"A"
027234  027272                        2$                      ;"B"
027236     340     340                .BYTE   340,340         ;PRIO. AT 7
027240  052777  121000  152114        BIS     #BIT15!BIT9!BIT13!BIT9,@DVSCR
027246  000240                        NOP                     ;SET BOTH INTERUPT RELATIVE BITS.
027250  104010                        HLT     10              ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
027252  004537  031654          1$:   JSR     R5,SETVEC       ;RESET VECTORS
027256  031676                        NO.ATRAP                ;"A"
```

```
                                    V..STRAP                           ;"S"
                                    .BYTE       340,340
                                    BIC         [illegible]            ;DSABLE DV11
                                    SCOPE                              ;SCOPE THIS TEST.
                        2$:         MOV         #STACK,SP              ;RESET STACK
                                    BIC         [illegible]            ;DSABLE DV11
                                    BR          1$                     ;RETURN
                        3$:         EMT         11                     ;VECTOR HERE WAS WRONG SIDE
                                    BR          3$

                                    ;************************* TEST 114 ********************************
                                    ;*TEST TO VERIFY THAT VECTOR "A"
                                    ;*OCCURES BEFOR VECTOR "B" EVEN
                                    ;*WHEN VECTOR "B" IS ENABLED BEFORE
                                    ;*VECTOR "A".
                                    ;;******************************************************************

                        ;   TEST 114
                        ;---------------
                        TST114:     MOV         #114,TSTNO
                                    MOV         #[illegible],NEXT
                                    MOV         #340,PS                ;SET CPU TO PRIO 7
                                    [illegible]                        ;RESET DV11
                                    CLR         R3                     ;SET COUNTER TO 0
                                    MOV         #2,R0                  ;WANT TO GET 2 INTERRUPTS
                                    JSR         R5,SETVEC              ;SET DV11 VECTORS
                                                                       ;RECEIVER INTERRUPTS TO 2$
                                                                       ;TRANSMITTER INTERRUPTS TO 3$
                                    .BYTE       340,340                ;SET PRIORITY TO 7 ON INTERRUPT
                                    MOV         #BIT9,DDVSCR           ;SET SYSTEM MAINT
                                    MOV         #BIT15+BIT13,DDVSCR
                                    MOV         #BIT9+BIT7+BIT6,DDVSCR
                                                                       ;SET B INTERRUPT ;A INTERRUPT ;CLEAR CPU STATUS
                                    [illegible] R0                     ;HANG WAITING FOR INTERRUPTS
                                                                       ;
                                    [illegible] R0                     ;DID TWO INTERRUPTS OCCUR
                                    BEQ         [illegible]            ;BR IF YES
                                    [illegible]                        ;EITHER NOT ENOUGH(2) OR TOO MANY (?2) INTERRUPTS
                        1$:         SCOPE                              ;SCOPE TEST
                        2$:         [illegible] R0                     ;RXISR
                                    CMP         #1,R0                  ;IF RX GOT HERE 1ST R0=1 & B=1
                                    BEQ         6$                     ;BR IF OK
                                    [illegible] 1                      ;RX NOT 1ST
                        6$:         [illegible] #BIT7+BIT6,DDVSCR      ;DISQUALIFY INTERRUPT REQST
                                    RTI                                ;RETURN
                        3$:         DEC         R0                     ;TX ISR
                                    BEQ         4$                     ;IF SECOND INTERRUPT R0 SHOULD NOW =0
                                    [illegible] 1$                     ;TX INTERRUPT OUT OF ORDER
                        4$:         RTI                                ;RETURN

                                    ;************************* TEST 115 ********************************
                                    ;*PRIORITY INTERUPT TESTS.
                                    ;*SET PS TO PRIORITY 7 AND VERIFY
                                    ;*THAT THE DV11 DOESN'T INTERUPT.
                                    ;;******************************************************************
```

```
                                        ;  TEST 115
                                        ;--------------
                                ST115:  MOV     #115,TSTNO
                                        MOV     #TST116,NEXT
                                        MOV     #340,PS                 ;LOCK OUT INTERUPTS
                                        MTCLR                           ;CLEAR ALL
                                        JSR     R5,SETVEC               ;PREPARE VECTORS
                                        .ATRAP                                  ;"4"
                                        .BTRAP                          ;"BPT"
                                        .BYTE   340,340                 ;PRIO. AT 7
                                        MOV     #BPT+BUSERR+BIT5,PC.USR
                                        MOV     #340,PS ;SET CPU PRIO AND ENABLE VECT. "4"
                                        CLR     @#4                     ;ZAP 1ST
                                        BR      2D.SCR                  ;ENABLE D.11
                                        .SCOPE                          ;SCOPE TEST
                                        MOV     #SP.,SP                 ;SET FOR RETURN
                                        ;

                                        ;************************ TEST 116 ****************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 5 AND VERIFY
                                        ;*THAT THE D.11 DOESN'T INTERUPT.
                                        ;;**********************************************************

                                        ;  TEST 116
                                        ;--------------
                                ST116:  MOV     #116,TSTNO
                                        MOV     #TST117,NEXT
                                        MOV     #340,PS                 ;LOCK OUT INTERUPTS
                                        MTCLR                           ;CLEAR ALL
                                        JSR     R5,SETVEC               ;PREPARE VECTORS
                                        .ATRAP                                  ;"4"
                                        .BTRAP                          ;"BPT"
                                        .BYTE   340,340                 ;PRIO. AT 7
                                        MOV     #BPT+BUSERR+BIT5,PC.USR
                                        MOV     #300,PS ;SET CPU PRIO AND ENABLE VECT. "4"
                                        CLR     @#4                     ;ZAP 1ST
                                        BR      2D.SCR                  ;ENABLE D.11
                                        .SCOPE                          ;SCOPE TEST
                                        MOV     #SP.,SP                 ;SET FOR RETURN
                                        ;

                                        ;************************ TEST 117 ****************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 5 AND VERIFY
                                        ;*THAT THE D.11 DOESN'T INTERUPT.
                                        ;;**********************************************************

                                        ;  TEST 117
                                        ;--------------
                                ST117:  MOV     #117,TSTNO
                                        MOV     #TST120,NEXT
                                        MOV     #340,PS                 ;LOCK OUT INTERUPTS
```

D09


```
                                   MSTCLR                  ;CLEAN DV11
        031654                     JSR     R5,SETVEC       ;PREPARE VECTORS
                                   NO.ATRAP                ;"A"
                                   NO.BTRAP                ;"B"
           340                     .BYTE   340,340         ;PRIO. AT 7
        001330  151466             MOV     #BIT9+BIT7+BIT6,@DVSCR
        000340  177776             MOV     #240,PS ;SET CPU PRIO AND ENABLE VECT. "A"
                                   NOP                     ;WAIT
        151452             1$:     CLR     @DVSCR          ;DSABLE DV11
                                   SCOPE                   ;SCOPE THIS TEST
        027704             2$:     MOV     #1$,(SP)        ;SET FOR RETURN
                                   RTI                     ;

                                   ;*********************** TEST 120 ******************************
                                   ;*PRIORITY INTERUPT TESTS.
                                   ;*SET PS TO PRIORITY 4 AND VERIFY
                                   ;*THAT THE DV11 DOES INTERUPT.
                                   ;;***************************************************************

                           ;   TEST 120
                           ;--------------
        000120  001226     TST120: MOV     #120,TSTNO
                                   MOV     #TST121,NEXT
        000340  177776             MOV     #340,PS         ;LOCK OUT INTERUPTS
                                   MSTCLR                  ;CLEAN DV11
        031654                     JSR     R5,SETVEC       ;PREPARE VECTORS
                                                           ;"A"
                                   NO.BTRAP                ;"B"
           340                     .BYTE   340,340         ;PRIO. AT 7
        001330  151376             MOV     #BIT9+BIT7+BIT6,@DVSCR
        000200  177776             MOV     #200,PS ;SET CPU PRIO AND ENABLE VECT. "A"
                                   NOP                     ;WAIT
                                   HLT     0               ;DV FAILED TO INTERUPT
        151360             1$:     CLR     @DVSCR          ;DSABLE DV11
                                   SCOPE                   ;SCOPE THIS TEST
        027776             2$:     MOV     #1$,(SP)        ;SET FOR RETURN
                                   RTI                     ;

                                   ;*********************** TEST 121 ******************************
                                   ;*TEST THAT BIT15,NPR STATUS INTR WILL
                                   ;*SET WHEN AN ENTRY IS MADE INTO THE
                                   ;*NPR STATUS REGISTER.
                                   ;*ALSO VERIFY THAT READING THE DVNSR CLEARS DVSCR BIT15.
                                   ;;***************************************************************

                           ;   TEST 121
                           ;--------------
        000121  001226     TST121: MOV     #121,TSTNO
        032114  001216             MOV     #TST122,NEXT
                                   MSTCLR                  ;RESET DV11
        151326                     TST     @DVSCR          ;MAKE SURE 15 =0
                                   BPL     64$             ;BR IF OK
                                   HLT     0               ;15 S B=0
        000010  151314     64$:    MOV     #BIT3,@DVSCR    ;SET SOURCE SEC
```

E09


```
                                                        MOV     @XFR+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,SD.SPC
                                                        CLR     R0                      ;XFR ALL RESULTS TO ZERO
                                                        CSYNCLK                         ;EXECUTE
                                                        BIS     #1,SD.SCR               ;GO!!
                                                        BMI     .-6                     ;BR IF YES
                                                        WAIT                            ;WASTE TIME
                                                        DELAY   2                       ;DO DELAY AND WAIT
                                                        .
                                                        .                               ;15 NEVER SET
                                                        MOV     SD.SCR,R0               ;READ NSR
                                                        BIT     #100000,R0              ;15 CLEAR IN SCR?
                                                        BNE     33$                     ;BR IF YES
                                                        .                               ;BIT NOT CLEARED BY READING NSR
                                                33$:    SCOPE                           ;SCOPE TEST


                                                        ;************************ TEST 122 ****************************
                                                        ;*TEST TO WRITE PATTERNS THROUGH
                                                        ;*LINE MAP STATUS REGISTERS.
                                                        ;*BITS WRITTEN: 11,10,9,8,03,02,01,00
                                                        ;* WHEN BIT IS OF 1'S BUT IS SET SO SHOULD BIT IS OF 0'S
                                                        ;**************************************************************

                                                ;  TEST 122
                                                ;--------------
                                                ST122:  MOV     #122.,TSTNO
                                                        MOV     #100,TSTTIM             ;1 SEC
                                                        MOV     #1,LOOPCNT              ;ONE PASS
                                                        CLR     SD.SCR                  ;RESET SYNC
                                                        BIS     #BIT3,SD.SCR            ;SET SCANNER TO BE
                                                        MOV     #0,R0                   ;ZERO LINE NUMBER POINTER
                                                        MOV     #TEMP0,R1               ;ZERO TEMP0
                                                        MOV     R0,(R1)                 ;SET POINTER
                                                        MOV     (R1),SD.SPC             ;
                                                        BIS     #BIT15,TEMP1            ;SET CORRESPONDING BIT TO BE FOUND IN MSR.
                                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT4,R2
                                                1$:     CSYNCLK                         ;XFR RAM OUTPUT TO MSR
                                                        BIT     SD.SCR                  ;WAIT FOR
                                                        .
                                                        MOV     TEMP1,R5                ;GET GOOD 15=1 DATA
                                                        MOV     SD.SCR,R4               ;READ MSR
                                                        BIC     R5,R4                   ;OK?
                                                        BEQ     3$                      ;BR IF GOOD
                                                        BIS     #3,ERRFLG               ;DVNSR HAS WRONG DATA
                                                        BR      4$                      ;LOOK ON CHIP
                                                3$:     PERFORM .GETSCAN                ;POINT MSCANNER
                                                        BIS     TEMP1,R0                ;SET ONE
                                                        MOV     R0,SD.SPC               ;POINT DATA
                                                        BIT     #BIT4,RO                ;ALL DONE?
                                                        BEQ     1$                      ;BR IF NO.
                                                        BIC     #0,BIT15,TEMP1          ;CLEAR ALL BUT ENTRY PRESENT
                                                        MOV     R0,(R1)                 ;ZERO LINE POINTER
                                                        MOV     R0,MSCANNER,POINTER     ;ZERO MSCANNER POINTER
                                                        MOV     TEMP0,SD.SPC            ;UPDATE HIGH BYTE DATA
                                                        MOV     TEMP0,TEMP1+1           ;SAVE IN CASE LOOP AGAIN
                                                        MOV     TEMP0,R2                ;ADDRESS
```

```
                                                                    ;RAM READ
                                                                    ;
                                                                    ;ALL DONE?
                                                                    ;GO IF NO
                                                                    ;SCOPE TEST.

                                                                    ;********** FIRST PASSED ATTEMPT *********
                                                                    ;**********      TO EXECUTE NPR.   *********
                                                                    ;*-----------------------------------------

                                                                    ;************************ TEST 123 ****************************
                                                                    ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                                                    ;*TEST THAT THE DV11 CAN "READ" FROM CORE LOCATION
                                                                    ;*0 IN THE NPR LOGIC.
                                                                    ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                                                    ;*PLACED INTO IT AND READ INTO THE DV11 AND XFERED
                                                                    ;*INTO THE SCRIC REGISTER.
                                                                    ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
                                                                    ;;**************************************************************

                                                                    ;  TEST 123
                                                                    ;--------------
                                                                    ST123: MOV     #123,TSTNO
                                                                           MOV     #ST124,NEXT
                                                                           MOV     #1$,LOCK
                                                                           RSTCLR                   ;RESET DV11
                                                                           MOV     #BIT3,@DVSCR     ;SET SOURCE SE..
                                                                           DVR1X,R3                 ;SET POINTERS
                                                                           DVSFR,R2                 ;
                                                                           MOV     #NPRLOC,@DVSRA
                                                                           MOV     #RAM,(R2)        ;RAM READ
                                                                           ROMCLK                   ;EXECUTE
                                                                           MOV     #XFR+BIT7+BIT5+3,(R2) ;BIT4+BIT1+BIT0
                                                                           ROMCLK                   ;DO XFR TO NPR ADD.
                                                                           CLR     NPRLOC           ;CLEAR NPR LOCATION
                                                                           CLR     R5               ;CLEAR R5=DATA
                                                                           CLR     @DVSRA           ;CLEAR DATA PORT
                                                                    1$:    MOV     #NPR,(R2)        ;DO THE NPR
                                                                           ROMCLK                   ;***NOW***
                                                                           MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
                                                                           ROMCLK                   ;XFR NPR OUTPUT TO D.RIC
                                                                           MOV     (R3),R4          ;READ RIC
                                                                           MOV     NPRLOC,R5 ;READ  ;ADD DATA
                                                                           CMP     R5,R4            ;OK?
                                                                           BEQ     2$               ;BR IF YES
                                                                           HLT     13               ;NPR FUNCTION FAILED
                                                                    2$:    SCOPE1                   ;LOOP?
                                                                           INCB    NPRLOC ;UPDATE DATA
                                                                           BNE     1$               ;BR IF MORE
                                                                           SCOPE                    ;SCOPE TEST

                                                                    ;************************ TEST 124 ****************************
                                                                    ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                                                    ;*TEST THAT THE DV11 CAN "READ" FROM CORE LOCATION
```

```
                                        ;*VIA THE NPR LOGIC
                                        ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                        ;*PLACED INTO IT AND READ INTO THE DVII AND XFERED
                                        ;*INTO THE DVPIC REGISTER.
                                        ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
                                        ;**********************************************************

                                ;  TEST 124
                                ;-------------
                        ST124:  MOV     #124,TSTNO
                                MOV     #ST124,NEXT
                                MOV     #10.,LOOP
                                MSTCLR                  ;RESET DVII
                                MOV     #NPRLOC+1,@#DVCSR       ;SET NPR ADDRESS
                                MOV     #NPRLOC,R0              ;
                                MOV     #NPRLOC+1,00.BRA
                                BRA     .+2             ;RAM READ
                                NOP
                                MOV     #XFR+BIT7+BIT5+9.,@#DVCSR       ;DO XFER TO NPR REG.
                                NOP
                                CLR     @R0             ;CLEAR NPR LOCATION
                                CLR     @#DVDBR         ;CLEAR GOOD
                                CLR     @#DVSRA         ;CLEAR DATA PORT
                                MOV     #NPR,.R2.       ;DO THE NPR
                                MSTCLR                  ;***NOW***
                                MOV     #XFR+BIT7+BIT4+3,@#DVCSR        ;XFER NPR OUTPUT TO DVPIC
                                NOP                     ;READ BIT
                                MOV     @R2,R4          ;GET GOOD DATA
                                CMP     @R0,R5          ;
                                BEQ     .+4             ;OK?
                                MOV     R4,@#DVDBR      ;SAVE IF YES
                                BR      ERR             ;NPR FUNCTION FAILED
                        10$:    SOB     R1,1$           ;LOOP
                                INC     @R0             ;UPDATE PATTERN
                                CMP     #4,@R0          ;DONE IF MORE
                                SCOPE                   ;SCOPE TEST

                                ;********************** TEST 125 ****************************
                                ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                ;*TEST THAT THE DVII CAN "WRITTEN" INTO CORE LOCATION
                                ;*VIA THE NPR LOGIC.
                                ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                ;*WRITTEN INTO IT BY THE DVII NPR LOGIC.
                                ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
                                ;**********************************************************

                                ;  TEST 125
                                ;-------------
                        ST125:  MOV     #125,TSTNO
                                MOV     #ST125,NEXT
                                MOV     #10.,LOOP
                                MSTCLR                  ;RESET DVII
```

```
                                                        MOV     #BIT3,DVSCR     ;SET SOURCE SEL.
                                                        MOV     DVRIC,R3        ;SET POINTERS
                                                        MOV     DVSFR,R2        ;
                                                        MOV     #NPRLOC+1,DVSRA
                                                        MOV     #RAM,(R2)       ;RAM READ
                                                        ROMCLK                  ;EXECUTE
                                                        MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
                                                        ROMCLK                  ;DO XFR TO NPR REG.
                                                        CLR     NPRLOC          ;CLEAR NPR LOCATION
                                                        CLR     R5              ;CLEAR GOOD
                                                        MOV     DVSRA,R4        ;WITH PORT
                                                        MOV     NPRLOC          ;CLEAR NPR LOC
                                                        MOV     #RAM,(R2)       ;DO RAM READ
                                                        ROMCLK                  ;
                                                        MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
                                                        ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
                                                        MOV     #NPR,(R2)       ;DO THE NPR
                                                        ROMCLK                  ;***NOW***
                                                        MOV     DVSRA,R5        ;READ GOOD DATA
                                                        MOV     NPRLOC,R4       ;READ DATA
                                                        BIT     #1,R4           ;OK?
                                                        BNE     .+6             ;BR IF YES
                                                        HALT                    ;NPR FUNCTION FAILED
                                                        BR      .               ;LOOP
                                                        CMP     DVSRA,R5        ;DUPLICATE DATA
                                                        BEQ     .+8             ;ALL DONE?
                                                        CMP     #^B1001,DVSRA   ;BR IF YES
                                                        BNE     .               ;WRONG
                                                        SCOPE                   ;SCOPE TEST
```

```
                                                ;*********************** TEST 126 ****************************
                                                ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                                ;*TEST THAT THE DV11 CAN "WRITTEN" INTO CORE LOCATION
                                                ;*VIA THE NPR LOGIC.
                                                ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                                ;*WRITTEN INTO IT BY THE DV11 NPR LOGIC.
                                                ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
                                                ;*************************************************************

                                        ;  TEST 126
                                        ;--------------
                                        ST126:  MOV     #126,TSTNO
                                                MOV     #TST127,NEXT
                                                MOV     #1$,LOOK
                                                MSTCLR                  ;RESET DV11
                                                MOV     #BIT3,DVSCR     ;SET SOURCE SEL.
                                                MOV     DVRIC,R3        ;SET POINTERS
                                                MOV     DVSFR,R2        ;
                                                MOV     #NPRLOC+1,DVSRA
                                                MOV     #RAM,(R2)       ;RAM READ
                                                ROMCLK                  ;EXECUTE
                                                MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
                                                ROMCLK                  ;DO XFR TO NPR REG.
                                                CLR     NPRLOC          ;CLEAR NPR LOCATION
                                                CLR     R5              ;CLEAR GOOD
```

```
                                                        ;CLEAR DATA PORT
                                                        ;CLEAR NPR LOC
                                                        ;DO RAM READ
                                                        ;
                                                        ;XFR RAM OUTPUT TO NPR INPUT DATA
                                                        ;DO THE NPR
                                                        ;***NOW***
                                                        ;READ GOOD DATA
                                                        ;GET DATA
                                                        ;
                                                        ;OK?
                                                        ;BR IF YES
                                                        ;NPR FUNCTION FAILED
                                                        ;LOOP?
                                                        ;UPDATE DATA
                                                        ;ALL DONE?
                                                        ;BR IF NO
                                                        ;SCOPE TEST
```

```
;********************** TEST 127 ****************************
;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
;*TEST THAT THE DZ11 CAN DO AN NPR
;*TO A NON-EXISTANT MEMORY.
;*TEST THAT BRANCH "A" -NXM H- IS SET AFTER
;*THE NPR. THEN DO A "SET/CLEAR" CLEAR NXM
;*AND VERIFY THAT IT IS CLEARED.
;************************************************************
```

```
;  TEST 127
;--------------
```

```
                                                        ;RESET DZ11
                                                        ;SET SOURCE SEL.
                                                        ;SET POINTERS
                                                        ;
                                                        ;SET EA BITS.
                                                        ;LOAD "NXM".
                                                        ;RAM READ
                                                        ;EXECUTE
                                                        ;EXECUTE
                                                        ;DO THE NPR
                                                        ;***NOW***
                                                        ;BRANCH "A" NXM
                                                        ;READ BRANCH TEST POINTS.
                                                        ;GET IMAGE
                                                        ;BR "A" TRUE
                                                        ;BR B FALSE
                                                        ;NXM TRUE?
                                                        ;BR IF YES
                                                        ;BR "A" OF NXM FAILED!
```

J09


```
                                                        ;SET/CLEAR  CLEAR NXM.
                                                        ;EXECUTE.
                                                                ;BRANCH "B" NXM
                                                        ;READ BRANCH TEST POINTS.
                                                        ;GET IMAGE
                                                        ;BR A FALSE  BR B FALSE
                                                        ;NXM TRUE
                                                        ;BR IF NO
                                                        ;BR "B" IF NXM FAILED TO CLEAR!
                                                        ;SCOPE TEST
```

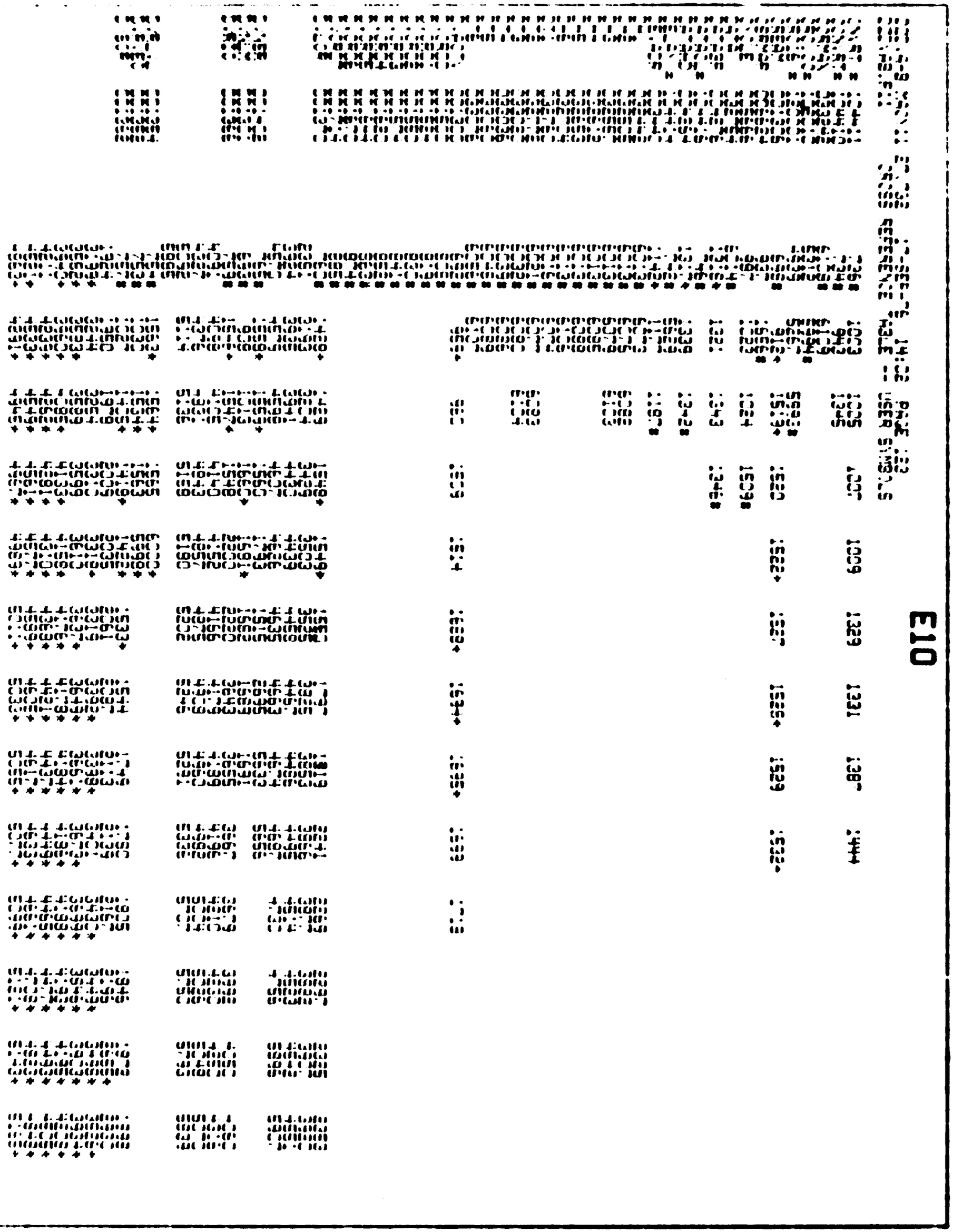
```
031342                                  SETSCAN:
031342  010346                                  MOV     R3,-(SP)
031344  052777  000010  150010                  BIS     #BIT3,@DVSCR
031350  012503                                  MOV     (R5)+,R3
031352  001414                                  BEQ     2$
031354  012777  050102  150014          1$:     MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
031362  104415                                  ROMCLK
031364  005201                                  INC     R1
031366  012777  050102  150002                  MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
031374  104415                                  ROMCLK
031376  005201                                  INC     R1
031400  005303                                  DEC     R3
031402  001364                                  BNE     1$
031404  012603                          2$:     MOV     (SP)+,R3
031406  010100                                  MOV     R1,R0
031410  000241                                  CLC
031412  006000                                  ROR     R0
031414  000205                                  EXIT

                                        ;SUBROUTINE TO LOAD DATA INTO "A" AND "B" REG.
                                        ;THE FIRST ARGUMENT WILL LOAD THE "A" REGISTER;
                                        ;THE SECOND ARGUMENT WILL LOAD THE "B" REGISTER;
                                        ;AND THE THIRD ARGUMENT WILL SPECIFY THE POLYNOMIAL USED.

031420  012377  147752          L.DATA: MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE "A" REG
031424  012710  020000                  MOV     #RAM,(R0)       ;DO A ROM READ INSTR.
031430  104415                          ROMCLK                  ;EXECUTE
031432  012710  030251                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
031436  104415                          ROMCLK                  ;DATAXFR FROM RAM OUTPUT TO "A" REGISTER
031440  012377  147732                  MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE "B" REG.
031444  012710  020000                  MOV     #RAM,(R0)       ;DO A ROM READ
031450  104415                          ROMCLK                  ;
031452  012710  030262                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
031456  104415                          ROMCLK                  ;DO A DATA XFER FROM RAM OUTPUT TO THE "B" REG.
031460  012377  147712                  MOV     (R3)+,@DVSRA    ;PLACE DATA TO REMAIN IN THE RAM OUTPUT REG
031464  012710  020000                  MOV     #RAM,(R0)       ;DO A RAM READ TO SPECIFY POLYN.
031470  104415                          ROMCLK                  ;READ
031472  000203                          RTS     R3              ;LEAVE HERE

031474  010046                  SIMBCC: MOV     R0,-(SP)
031476  010146                          MOV     R1,-(SP)
031500  010246                          MOV     R2,-(SP)
031502  012537  001246                  MOV     (R5)+,TEMP1
031506  012537  001250                  MOV     (R5)+,TEMP2
031512  012537  001252                  MOV     (R5)+,TEMP3
031516  005037  031650          1$:     CLR     BCCFBK
031522  013700  001252                  MOV     TEMP3,R0
031526  006037  001250                  ROR     TEMP2
031532  005500                          ADC     R0
031534  032700  000001                  BIT     #BIT0,R0
031540  001402                          BEQ     2$
031542  005137  031650                  COM     BCCFBK
031546  013700  031646          2$:     MOV     XPOLY,R0
031552  005100                          COM     R0
```

```
031554  040037  031650                 BIC     R0,BCCFBK
031556  000241                          CLC
031560  006300                          ROR     TEMP3
031562  013700  031650                 MOV     BCCFBK,R0
031566  013701  001252                 MOV     TEMP3,R1
031572  010102                          MOV     R1,R2
031574  040100                          BIC     R1,R0
031576  042702  031650                 BIC     BCCFBK,R2
032002  060200                          BIS     R2,R0
032004  043737  031646  001252         BIC     XPOLY,TEMP3
032012  050037  001252                 BIS     R0,TEMP3
032016  005337  001246                 DEC     TEMP1
032022  001363                          BNE     1$
032024  013737  001252  031652         MOV     TEMP3,CALBCC
032032  012602                          MOV     (SP)+,R2
032034  012601                          MOV     (SP)+,R1
032036  012600                          MOV     (SP)+,R0
032040  000205                          RTS     R5

032042  000000                  XPOLY:  0
032044  000000                  BCCFBK: 0
032046  000000                  CALBCC: 0
                                CRC8=200
                                CRC16=120001
                                CRC.CCITT=102010

032050                          SETVEC:
032050  012577  147472                  MOV     (R5)+,@DVRVEC
032054  012577  147472                  MOV     (R5)+,@DVTVEC
032060  112577  147464                  MOVB    (R5)+,@DVRLVL
032064  112577  147464                  MOVB    (R5)+,@DVTLVL
032070  000205                          RTS     R5

032072                          NO.ATRAP:
032072  104011                          HLT     11
032074  000002                          RTI

032076                          NO.BTRAP:
032076  104012                          HLT     12
032100  000002                          RTI
```

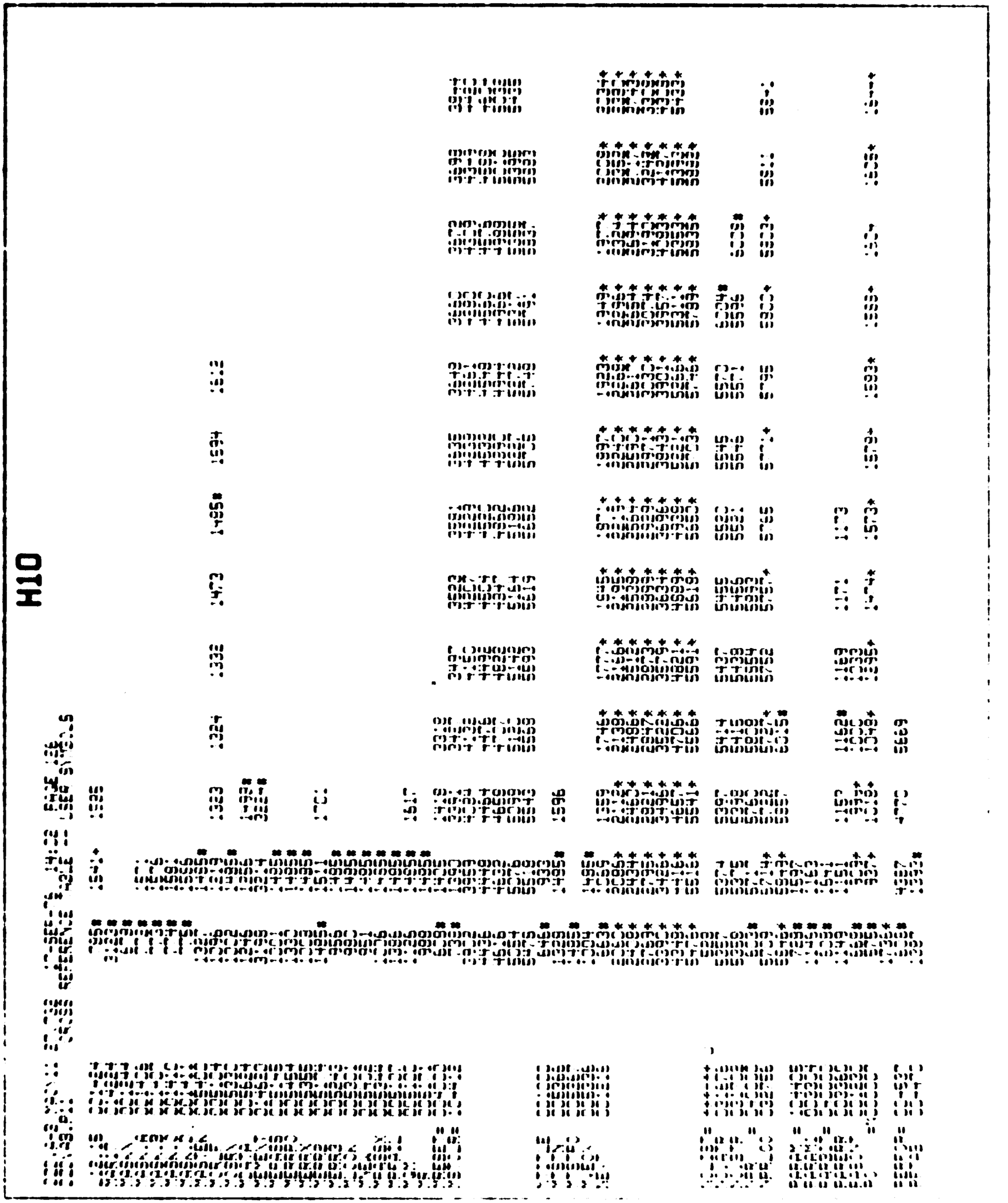
```
6013                                            .NLIST  BEX
        032726  050377  044522  040515  EM1:    .ASCIZ  <377>/PRIMARY REGISTER ADDRESSING TIME-OUT/
        033154  051777  041505  047117  EM2:    .ASCIZ  <377>!SECONDARY REGISTER READ/WRITE TEST!
        033020  050377  044522  040515  EM3:    .ASCIZ  <377>!PRIMARY REGISTER READ/WRITE TEST!
        033062  046777  046505  051117  EM4:    .ASCIZ  <377>!MEMORY EXTENSION READ/WRITE TEST!
        033124  051777  042520  044503  EM5:    .ASCIZ  <377>/SPECIAL FUNCTION REG TEST/
        033157     377  042526  052103  EM6:    .ASCIZ  <377>/VECTOR "A" FAILED TO INTERUPT/
        033216  053377  041505  047524  EM7:    .ASCIZ  <377>/VECTOR "B" FAILED TO INTERUPT/
        033255     377  047125  054105  EM8:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "A"
        033320  052777  042516  050130  EM9:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "B"
        033363     377  051120  046511  EM10:   .ASCIZ  <377>/PRIMARY REGISTER ERROR/
        033413     377  044514  042516  EM11:   .ASCIZ  <377>/LINE CARD STATIC TEST/
        033442  051377  043505  051511  DH1:    .ASCIZ  <377>/REGISTER REFERENCED  TRAPPED FROM/
        033505     377  054105  042520  DH2:    .ASCIZ  <377>/EXPECTED  FOUND   LINE  SEC REG  PRI REG/
        033557     377  054105  042520  DH3:    .ASCIZ  <377>/EXPECTED  FOUND   PRI REG/
        033612  045377  051123  050040  DH4:    .ASCIZ  <377>/JSR PC  DVSFR   EXPECTED  FOUND
        033663     377  053104  043123  DH5:    .ASCIZ  <377>/DVSFR   EXPECTED  FOUND/
        033734  046777  052123  041523  DH6:    .ASCIZ  <377>/MSTSCAN  DVSFR  EXPECTED  FOUND  LINE/
                032754                  .EVEN
                000000                  SKIP=000000
6014    032754  000000                  DATA:   0
6015    032756  000000                  NPRLOC: 0
6016    032760  000002                  DT1:    2
6017    032762     006  017                     .BYTE   6,15.
6018    032764  001262                  SAVR1
6019    032766     006  002                     .BYTE   6,2
6020    032770  001264                  SAVR2
6021    032772  000005                  DT2:    5
6022    032774     006  004                     .BYTE   6,4
6023    032776  001270                  SAVR4
6024    033000     006  002                     .BYTE   6,2
6025    033002  001266                  SAVR3
6026    033004     002  004                     .BYTE   2,4
6027    033006  001260                  SAVR0
6028    033010     002  007                     .BYTE   2,7
6029    033012  001262                  SAVR1
6030    033014     006  002                     .BYTE   6,2
6031    033016  001264                  SAVR2
6032    033020  000003                  DT3:    3
6033    033022     006  004                     .BYTE   6,4
6034    033024  001272                  SAVR5
6035    033026     006  002                     .BYTE   6,2
6036    033030  001270                  SAVR4
6037    033032     006  002                     .BYTE   6,2
6038    033034  001266                  SAVR3
6039    033036  000004                  DT4:    4
6040    033040     006  002                     .BYTE   6,2
6041    033042  001262                  SAVR1
6042    033044     006  002                     .BYTE   6,2
6043    033046  001266                  SAVR3
6044    033050     006  004                     .BYTE   6,4
6045    033052  001272                  SAVR5
6046    033054     006  001                     .BYTE   6,1
6047    033056  001270                  SAVR4
6048    033060                          DT5:
6049    033060  000003                  3
```

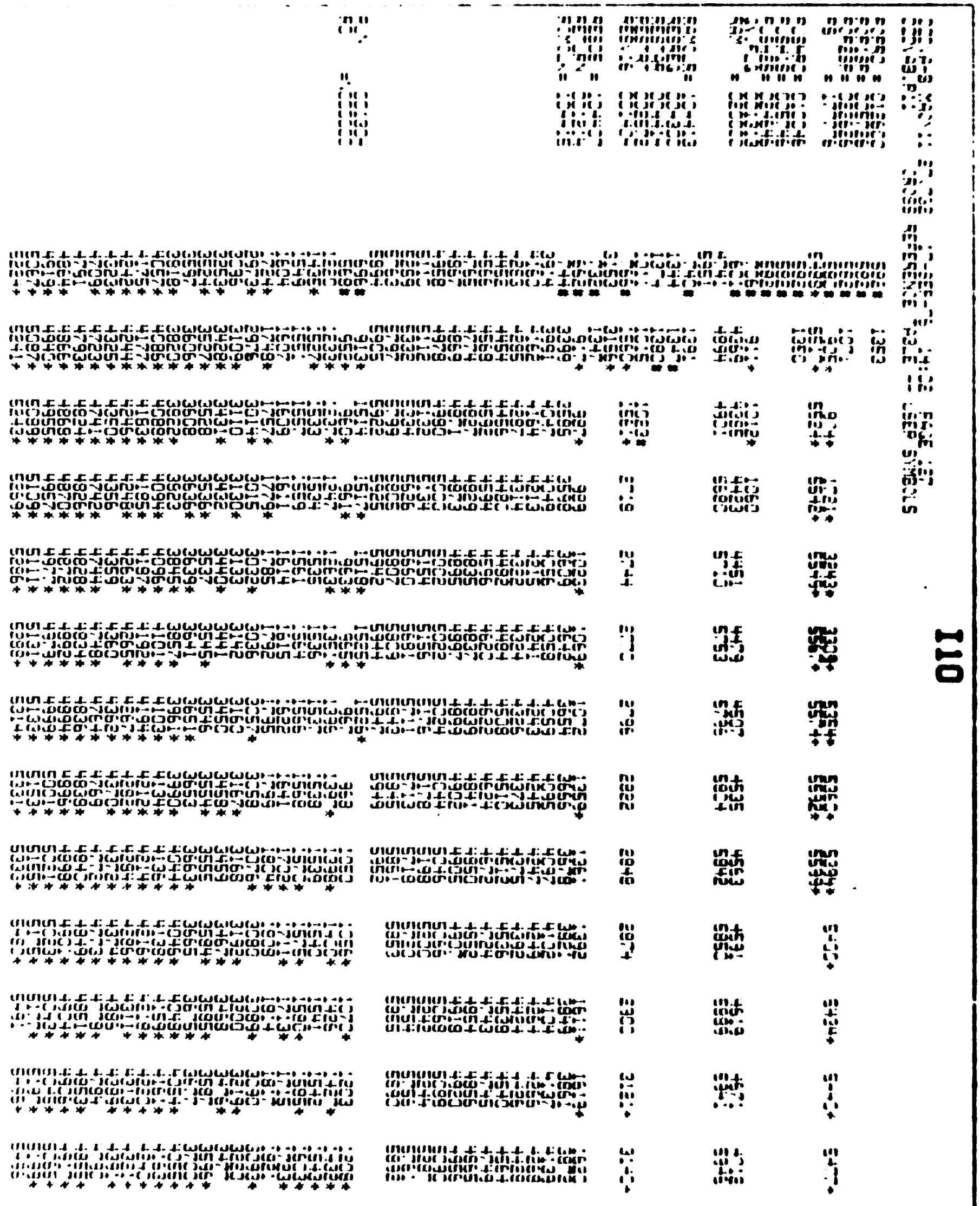
```
 6050  033062  000006  002                .BYTE   6.2
 6051  033064  001264                     SAVR2
 6052  033066  000006  004                .BYTE   6.4
 6053  033070  001272                     SAVR5
 6054  033072  000006  001                .BYTE   6.1
 6055  033074  001270                     SAVR4
 6056  033076  000005              DT6:   5
 6057  033100  000006  003                .BYTE   6.3
 6058  033102  001260                     SAVR0
 6059  033104  000006  001                .BYTE   6.1
 6060  033106  001264                     SAVR2
 6061  033110  000006  004                .BYTE   6.4
 6062  033112  001272                     SAVR5
 6063  033114  000006  001                .BYTE   6.1
 6064  033116  001270                     SAVR4
 6065  033120  000002  001                .BYTE   2.1
 6066  033122  001262                     SAVR1
 6067
 6068  033124                    .ERRTAB:
 6069  033124  000000                     0
 6070  033126  000000                     0
 6071  033130  000000                     0
 6072  033132  031706                     EM1
 6073  033134  032442                     DH1     ;HALT 1
 6074  033136  032760                     DT1
 6075  033140  031754                     EM2
 6076  033142  032505                     DH2     ;HALT 2
 6077  033144  032772                     DT2
 6078  033146  032020                     EM3
 6079  033150  032557                     DH3     ;HALT 3
 6080  033152  033020                     DT3
 6081  033154  032062                     EM4
 6082  033156  032505                     DH2     ;HALT 4
 6083  033160  032772                     DT2
 6084  033162  032124                     EM5
 6085  033164  032612                     DH4     ;HALT 5
 6086  033166  033036                     DT4
 6087  033170  032124                     EM5
 6088  033172  032653                     DH5     ;HALT 6
 6089  033174  033060                     DT5
 6090  033176  032157                     EM6
 6091  033200  000000                     0       ;HALT   7
 6092  033202  000000                     0
 6093  033204  032216                     EM7
 6094  033206  000000                     0       ;HALT   10
 6095  033210  000000                     0
 6096  033212  032255                     EM8
 6097  033214  000000                     0       ;HALT   11
 6098  033216  000000                     0
 6099  033220  032320                     EM9
 6100  033222  000000                     0       ;HALT 12
 6101  033224  000000                     0
 6102  033226  032363                     EM10
 6103  033230  032557                     DH3     ;HALT 13
 6104  033232  033020                     DT3
 6105  033234  032413                     EM11
```

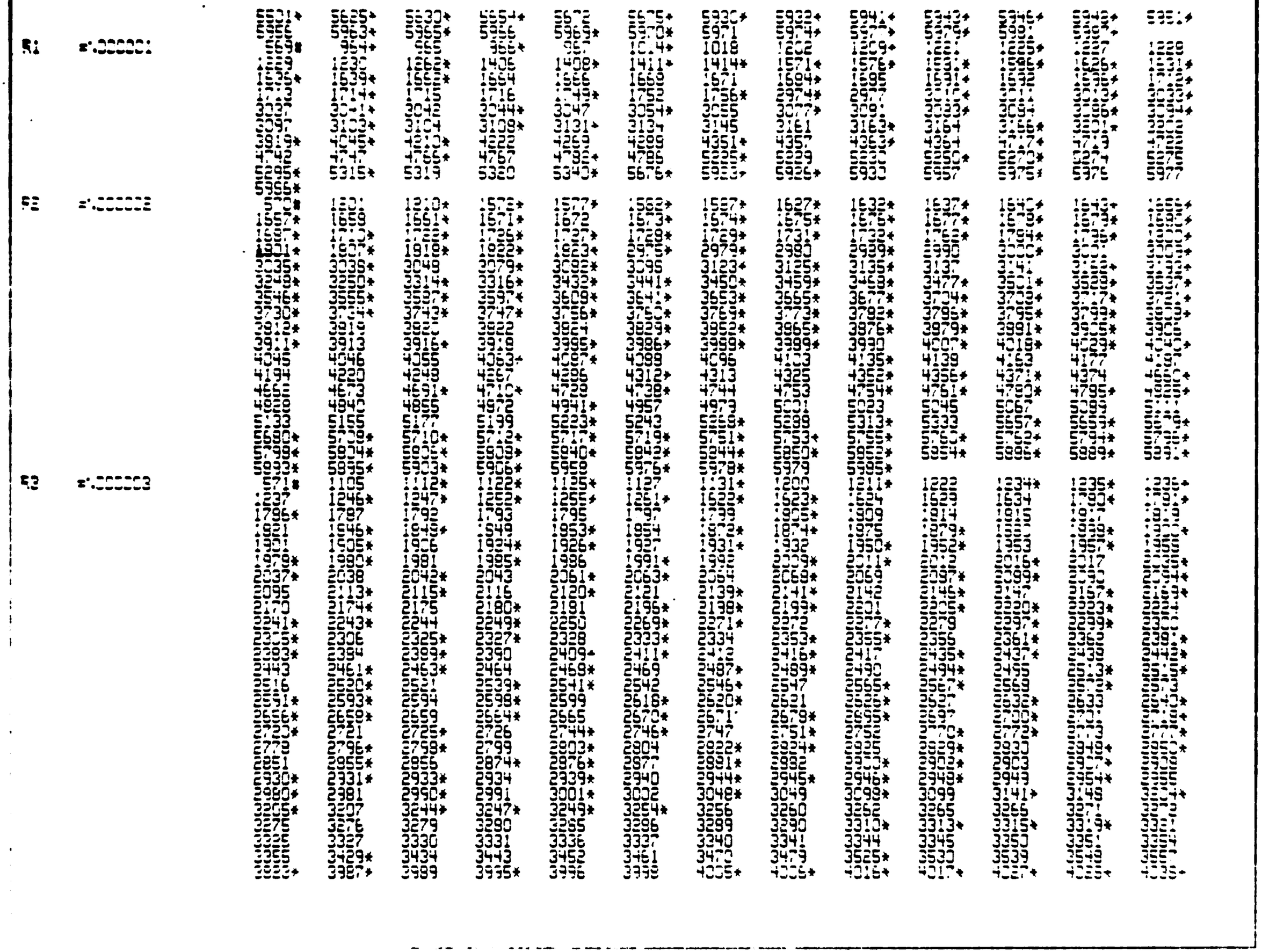
D10

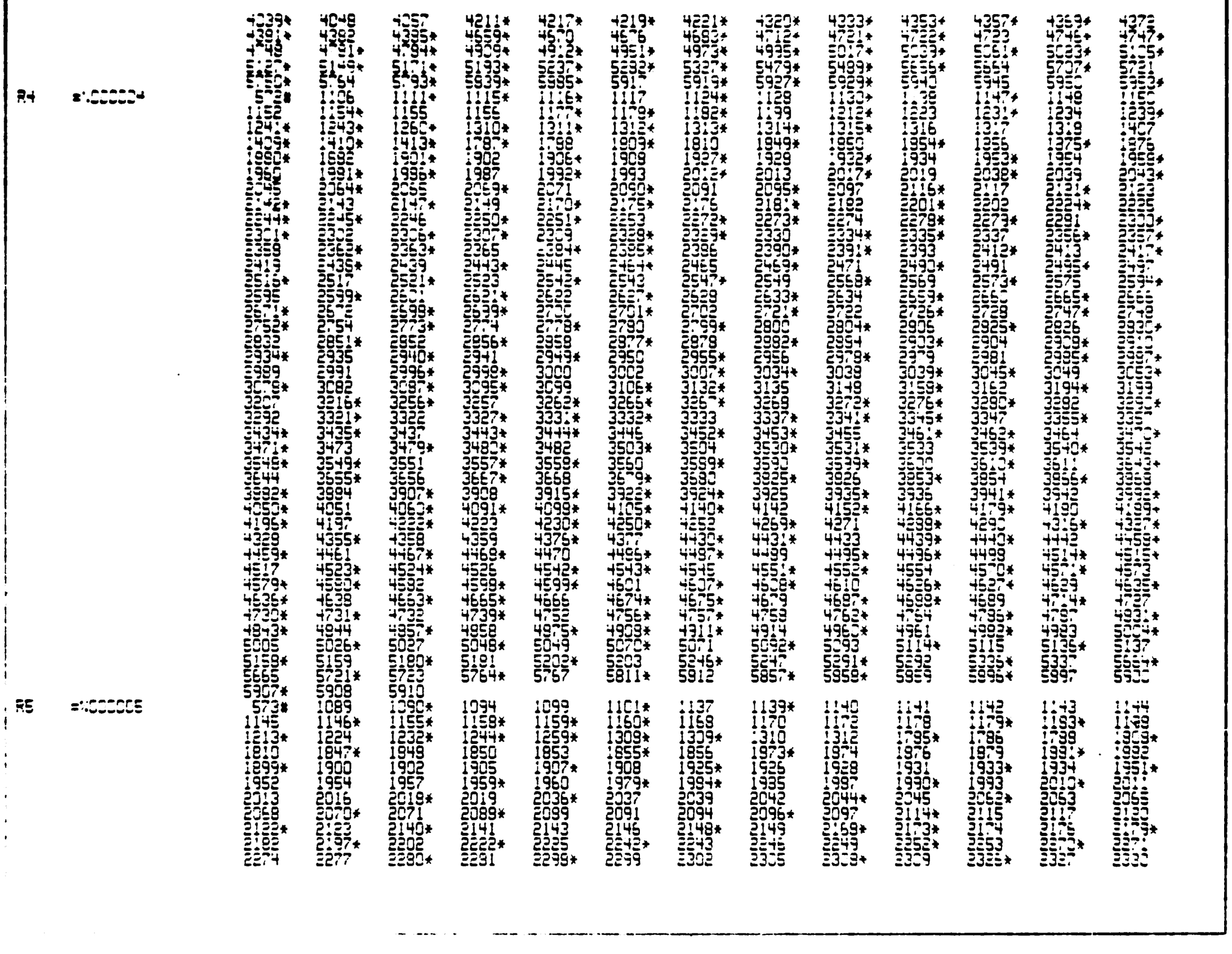


F10





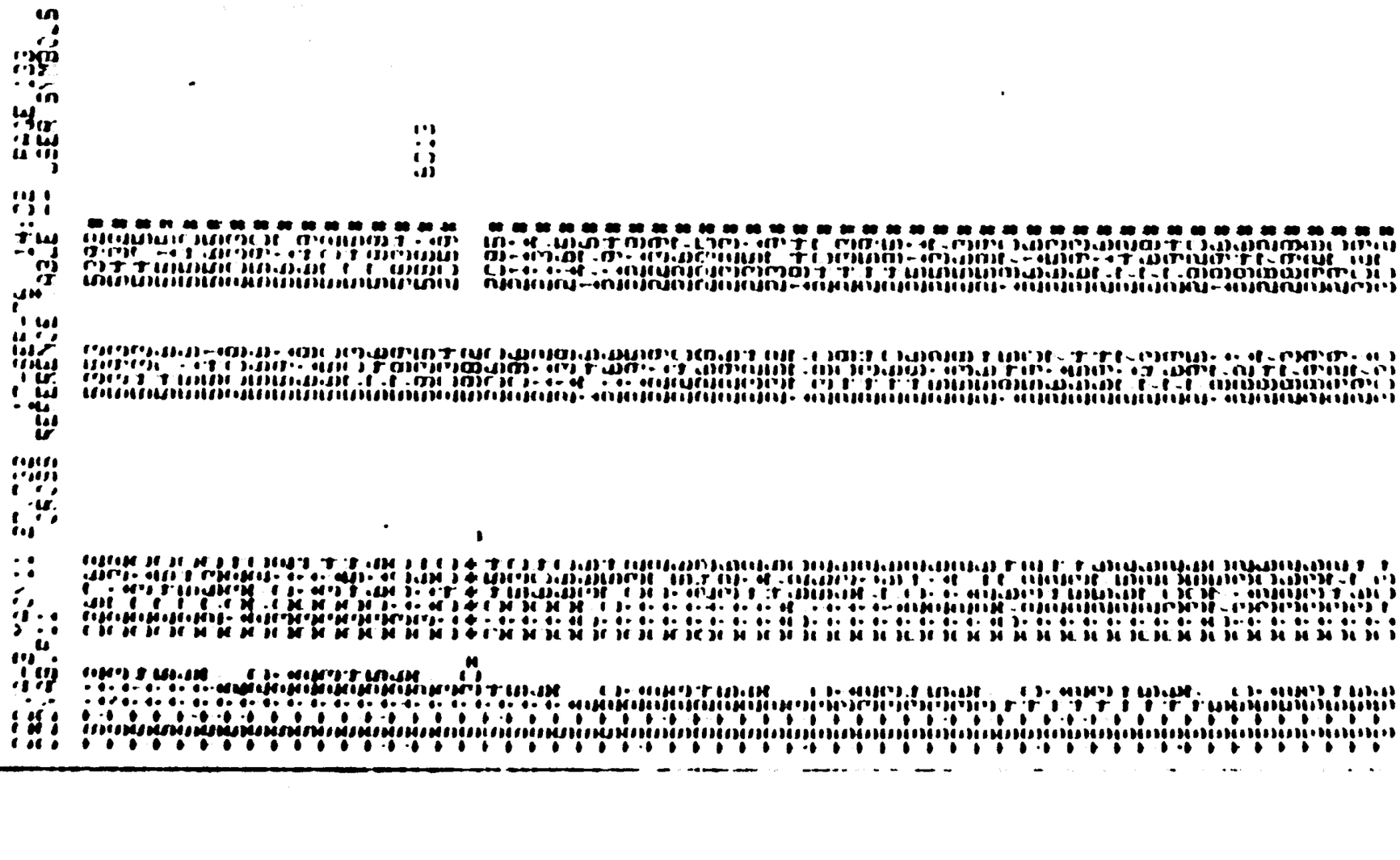


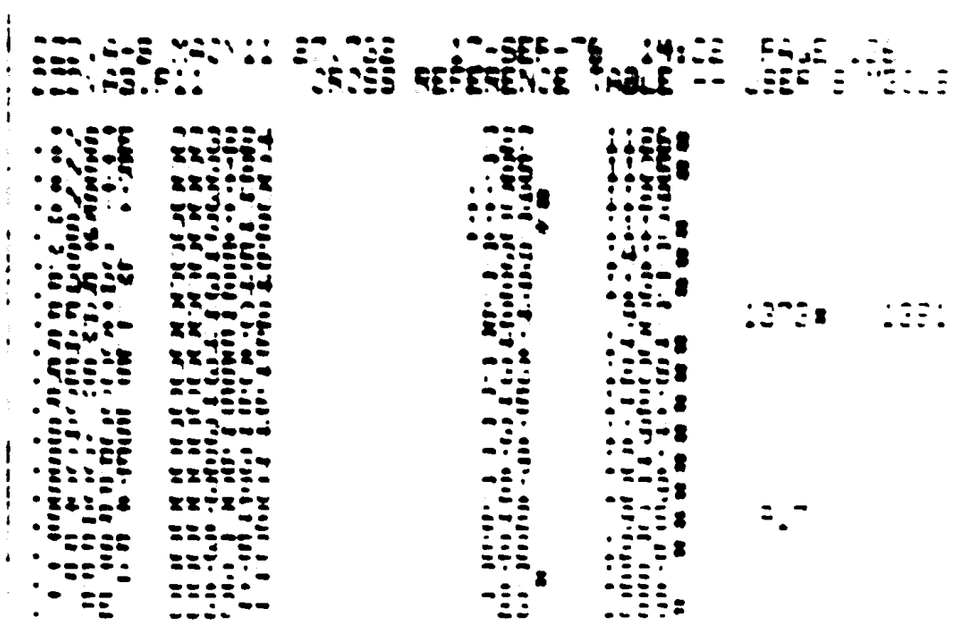


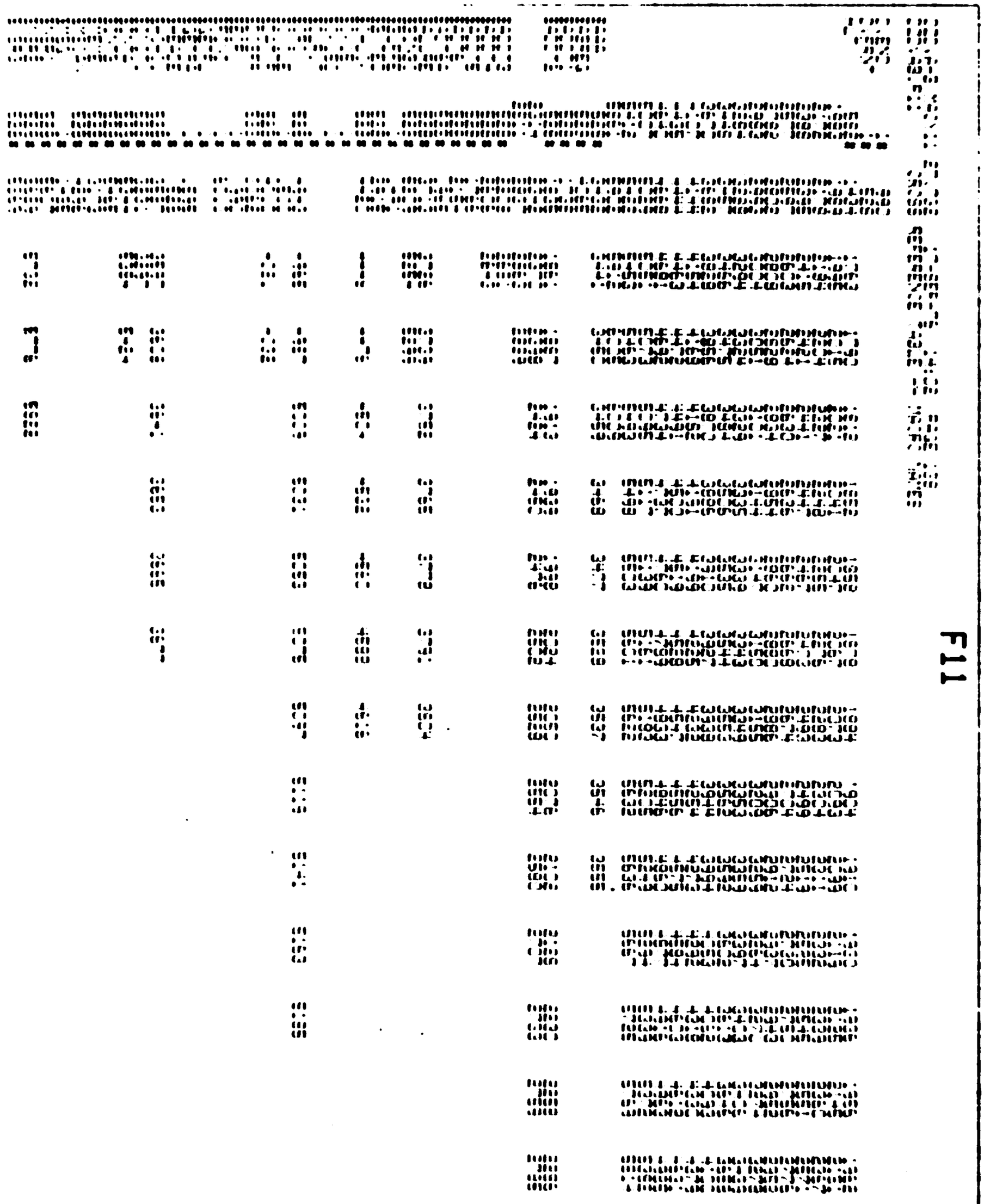
```
SETVEC  031664          5356    5366    5373    5396    5406    5413    5436    5447    5454    5481    5519    5544    5569
                        5894    5997#
SIMBOL  031474          4947    4969    4991    5013    5035    5057    5079    5101    5123    5145    5167    5189    5233
                        5279    5323    5956#
SKIP  = 000000          1828    1832    5013#
SP    = 000006           574#    906#    921#    922#    928     931     932     975#   1044#   1045#   1046    1050#   1070
                        1083#   1089#   1090    1091#   1101    1105#   1106#   1107    1108#   1124    1125    1127#   1128#
                        1130    1131    1137#   1138#   1139    1145#   1182    1183    1194    1220#   1221#   1222#   1223#
                        1224#   1225    1226#   1259    1260    1261    1262    1263    1276#   1277#   1278#   1279#   1280#
                        1281#   1282#   1283    1291#   1292#   1293    1303    1305    1309    1351    1360#   1382#   1396#
                        1400    1405#   1407#   1413    1414    1426#   1435    1448#   1449#   1450    1455    1461    1463
                        1465#   1466    1469#   1470#   1471    1705#   1731    1733    1734#   1762    4386#   4389    4726#
                        4727#   4729#   4738    4739    4740    4751#   4752#   4753#   4761    4762    4763    4769#   4773
                        5379#   5419#   5460#   5528#   5553#   5579#   5604#   5917#   5929    5956#   5957#   5958#   5985
                        5986    5987
SPACNT= 003745          1229#   1250    1253#   1267#
SPLR    000176           639#    930
STACK = 001200           581#    905     975     1360    1382    5379    5419    5460
START   001236           575#
SWDEF   003453          1198#
SWR     001203           650#    925     930#    935     949     954     959     978    1042    1051    1058    1079    1082
                        1289    1296    1301    1348    1355    1357    1441   1442#   1452#   1457#   1458#   1459#   1466#
                        1589
SW00  = 000001           561#    949
SW01  = 000002           560#   1589
SW02  = 000004           559#
SW03  = 000010           558#
SW04  = 000020           557#
SW05  = 000040           556#
SW06  = 000100           555#
SW07  = 000200           554#
SW08  = 000400           553#   1355
SW09  = 001000           552#   1079
SW10  = 002000           551#   1357
SW11  = 004000           550#   1058
SW12  = 010000           549#   1092    1296
SW13  = 020000           548#   1301
SW14  = 040000           547#
SW15  = 100000           546#
SYNA00  001505           811#
SYNA01  001532           822#
SYNA02  001556           833#
SYNA03  001602           844#
SYNA04  001626           855#
SYNA05  001652           866#
SYNA06  001675           877#
SYNA07  001722           888#
SYNB00  001512           813#
SYNB01  001536           824#
SYNB02  001562           835#
SYNB03  001606           846#
SYNB04  001632           857#
SYNB05  001656           868#
SYNB06  001702           879#
SYNB07  001726           890#
SYNCX   001240           576#
```

N10

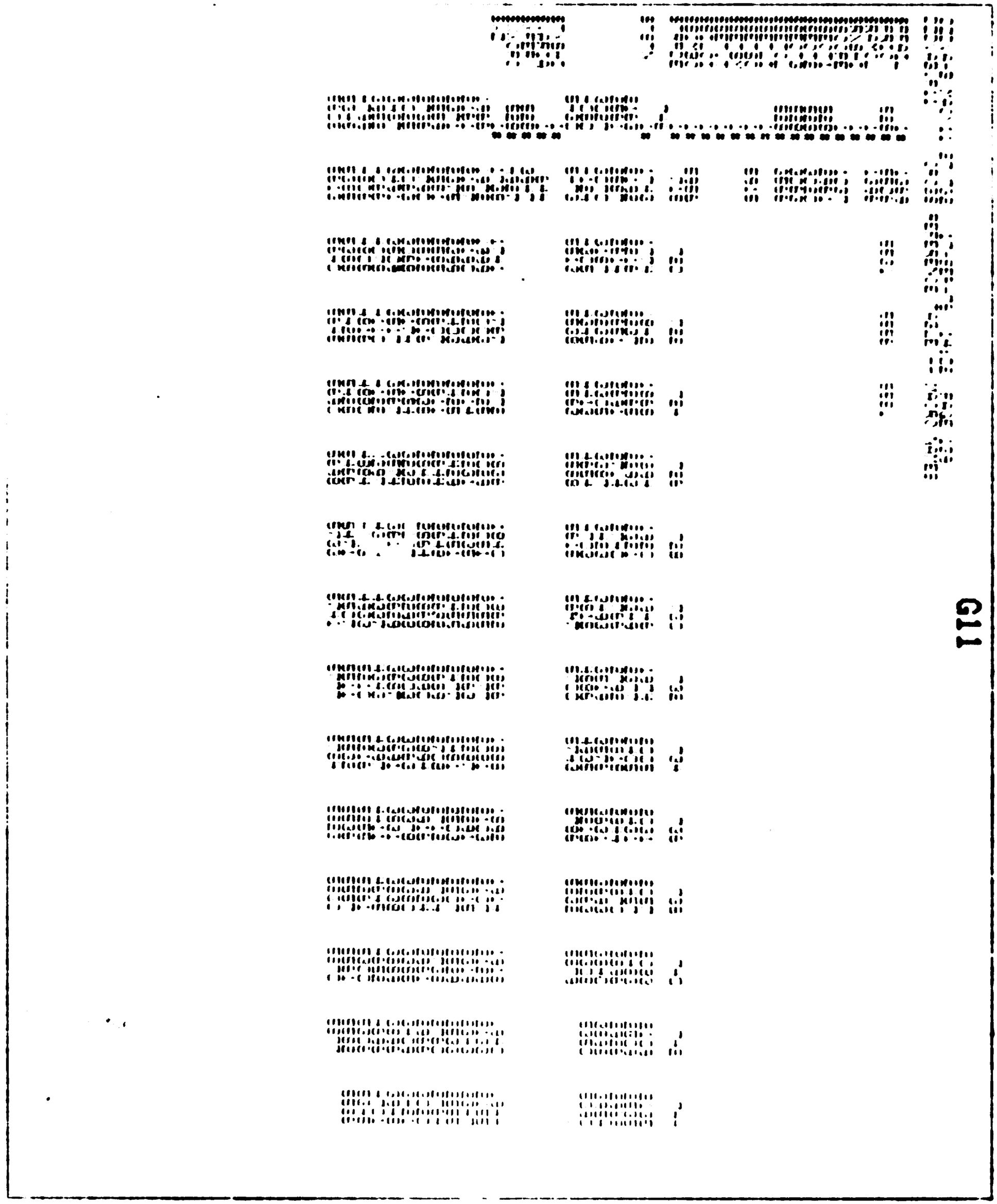

```
SYNC00  001516          815*
SYNC01  001542          826*
SYNC02  001566          837*
SYNC03  001612          848*
SYNC04  001636          859*
SYNC05  001662          870*
SYNC06  001706          881*
SYNC07  001732          892*
SYNC08  001420          787*    1536*
SYNC08  001430          788*    1538*
SYNC0C  001432          789*    1540*
SYNC0D  001434          790*    1542*
SYNC0D  001522          817*
SYNC0E  001546          828*
SYNC0F  001572          839*
SYNC00  001616          850*
SYNC04  001642          861*
SYNC05  001666          872*
SYNC06  001712          883*
SYNC07  001736          894*
S.C     = 050000        604*    3393    3585    3596    3606    3639    3651    3663    3675    3818    3848    3859    3871
                       3911     4044    4054    4265    4294    4321    4323    5903
TEMP    005562         1233     1383*   1384*   1496#
TEMP1   001246          679*     939*    940     945*    1487    1718*   1719*   3196*   3217*   4779*   4785    4794*   4795
                       5659*    5663    5671*   5674*   5679*   5959*   5932*
TEMP2   001250          680*     940*    941     1489    5655*   5677*   5678    5679    5682    5960*   5964*
TEMP3   001252          681*    5961*   5963    5973*   5975    5980*   5981*   5984
TEMP4   001254          682*
TEMP5   001256          683*    3198*   3201    3211*   3214*
TKCSR   001204          654*    1054    1113    1439    1446    1467
TKDBR   001206          655*    1044    1056    1115    1121    1291    1448    1469
TLAST   = 031164       1615     6013#
TPCSR   001210          656*    1097    1119    1298    1453
TPDBR   001212          657*    1099*   1121*   1300*   1455*
TRPOK   003762         1279#
TSTNO   001226          667*     916*   1368    1395    1600    1607    1609    1745*   1776*   1844*   1870*   1996*   1923*
                       1948*    1976*   2007*   2033*   2059*   2085*   2111*   2137*   2165*   2194*   2218*   2239*   2267*
                       2295*    2323*   2351*   2379*   2407*   2433*   2459*   2485*   2511*   2537*   2563*   2589*   2616*
                       2653*    2693*   2716*   2742*   2768*   2794*   2820*   2846*   2872*   2998*   2928*   2970*   3033*
                       3072*    3189*   3241*   3307*   3389*   3425*   3435*   3521*   3578*   3633*   3695*   3842*   3900*
                       3956*    4092*   4126*   4307*   4347*   4415*   4656*   4705*   4820*   4888*   4937*   5220*   5265*
                       5310*    5352*   5392*   5432*   5475*   5515*   5540*   5565*   5590*   5617*   5649*   5723*   5771*
                       5788*    5834*   5881*
TST1    007256         1603     1619    1745#
TST10   010154         1949     1976#
TST100  021560         4348     4415#
TST101  023162         4416     4656#
TST102  023362         4657     4705#
TST103  023776         4706     4820#
TST104  024252         4821     4888#
TST105  024444         4889     4937#
TST106  026100         4938     5220#
TST107  026256         5221     5265#
TST11   010250         1977     2007#
TST110  026434         5266     5310#
TST111  026612         5311     5352#
```







F11



J11

[illegible]

```
                5723    5727    5729    5733    5741    5746    5760    5765    5771    5773    5777    5784    5789    5803    5811
                5815    5816    5823    5830    5835    5849    5857    5863    5870    5877    5882    6109
.IFF            1739    1741    1746    1747    1768    1772    1777    1778    1837    1840    1845    1846    1863    1866    1871
                1875    1889    1892    1897    1898    1915    1918    1923    1924    1941    1944    1949    1950    1967    1973
                1977    1978    2000    2003    2008    2009    2026    2029    2034    2035    2052    2055    2060    2061    2078
                2081    2086    2087    2104    2107    2112    2113    2130    2133    2138    2139    2156    2161    2156    2157
                2189    2190    2195    2196    2210    2214    2215    2219    2220    2222    2232    2235    2240    2241    2260
                2263    2268    2269    2288    2291    2296    2297    2316    2319    2324    2325    2344    2347    2352    2353
                2372    2375    2380    2381    2400    2403    2408    2409    2426    2429    2434    2435    2452    2455    2456
                2461    2478    2481    2486    2487    2504    2507    2512    2513    2530    2533    2538    2539    2556    2559
                2564    2565    2582    2585    2590    2591    2608    2611    2617    2618    2645    2648    2654    2655    2662
                2683    2689    2694    2695    2701    2709    2712    2717    2719    2735    2738    2743    2744    2761    2764
                2769    2770    2787    2790    2795    2796    2813    2816    2821    2822    2839    2842    2847    2848    2865
                2868    2873    2874    2891    2894    2899    2900    2917    2924    2929    2930    2962    2966    2971    2972
                3021    3025    3030    3031    3064    3069    3074    3075    3175    3184    3190    3191    3233    3237    3242
                3243    3254    3299    3303    3308    3309    3318    3364    3378    3381    3385    3390    3391    3410    3421
                3426    3427    3433    3442    3452    3461    3470    3479    3489    3491    3496    3497    3510    3517    3522
                3523    3530    3539    3548    3557    3568    3574    3579    3580    3619    3629    3634    3635    3688    3691
                3696    3697    3832    3838    3843    3844    3890    3896    3901    3902    3949    3952    3957    3959    4067
                4078    4083    4094    4111    4122    4127    4128    4298    4303    4308    4309    4338    4343    4348    4349
                4398    4411    4416    4417    4646    4652    4657    4658    4697    4701    4706    4707    4800    4806    4809
                4816    4821    4822    4881    4884    4889    4890    4929    4933    4938    4939    5211    5216    5221    5222
                5256    5261    5266    5267    5301    5306    5311    5312    5346    5348    5353    5354    5386    5389    5393
                5394    5426    5428    5433    5434    5467    5471    5476    5477    5509    5511    5516    5517    5533    5536
                5541    5542    5558    5561    5566    5567    5593    5586    5591    5592    5609    5613    5618    5619    5641
                5645    5650    5651    5691    5698    5703    5704    5722    5723    5734    5741    5746    5747    5765    5767
                5779    5784    5789    5790    5803    5812    5824    5830    5835    5836    5949    5857    5871    5877    5882
.IFT             528    1738    1741    1767    1772    1836    1840    1862    1866    1998    1992    1314    1918    1940    1944
                1956    1972    1999    2003    2025    2029    2051    2055    2077    2081    2103    2107    2129    2133    2155
                2161    2187    2190    2209    2214    2222    2231    2235    2259    2263    2287    2291    2315    2319    2343
                2347    2371    2375    2399    2403    2425    2429    2451    2455    2477    2481    2503    2507    2529    2533
                2555    2559    2581    2585    2607    2611    2644    2648    2682    2689    2697    2708    2712    2734    2738
                2750    2754    2786    2790    2812    2816    2838    2842    2864    2868    2990    2994    2916    2924    2951
                2966    3020    3025    3063    3068    3174    3184    3232    3237    3252    3298    3303    3318    3363    3379
                3380    3385    3409    3421    3433    3442    3451    3460    3469    3478    3487    3491    3509    3517    3529
                3538    3547    3556    3567    3574    3618    3629    3687    3691    3831    3838    3889    3896    3948    3952
                4066    4078    4110    4122    4297    4303    4337    4343    4397    4411    4645    4552    4696    4701    4799
                4806    4808    4816    4880    4884    4928    4933    5210    5216    5256    5261    5300    5306    5345    5348
                5385    5388    5425    5428    5466    5471    5507    5511    5532    5536    5557    5561    5592    5596    5603
                5613    5640    5645    5690    5698    5717    5723    5733    5741    5760    5765    5777    5784    5803    5812
                5823    5830    5843    5857    5870    5877
.IFTF            528
.IIF            1746    1747    1777    1778    1845    1846    1848    1850    1855    1871    1872    1974    1975    1981    1987
                1988    1900    1902    1907    1923    1924    1926    1928    1933    1949    1950    1952    1954    1959    1977
                1978    2008    2009    2011    2013    2018    2034    2035    2037    2039    2044    2060    2061    2063    2065
                2070    2086    2087    2089    2091    2096    2112    2113    2115    2117    2122    2138    2139    2141    2143
                2148    2166    2167    2195    2196    2219    2220    2240    2241    2243    2245    2251    2269    2269    2271
                2273    2279    2296    2297    2299    2301    2307    2324    2325    2327    2329    2335    2352    2353    2355
                2357    2363    2380    2381    2383    2385    2391    2408    2409    2411    2413    2418    2434    2435    2437
                2439    2444    2460    2461    2463    2465    2470    2486    2487    2489    2491    2496    2512    2513    2515
                2517    2522    2538    2539    2541    2543    2548    2564    2565    2567    2569    2574    2590    2591    2593
                2595    2600    2617    2618    2654    2655    2694    2695    2717    2718    2720    2722    2727    2743    2744
                2746    2748    2753    2769    2770    2772    2774    2779    2795    2796    2798    2800    2805    2821    2822
                2824    2826    2831    2847    2848    2850    2852    2957    2873    2974    2976    2979    2993    2999    2900
                2902    2904    2909    2929    2930    2971    2972    3030    3031    3073    3075    3190    3191    3242    3243
                3286    3209    3390    3391    3426    3427    3496    3497    3522    3523    3579    3580    3589    3590    3593
```

N11


```
                3600    3610    3611    3634    3635    3643    3655    3667    3679    3695    3697    3843    3844    3901    3922
                3956    3958    4083    4084    4128    4128    4308    4309    4348    4349    4416    4417    4657    4658    4704
                4820    4821    4822    4889    4890    4938    4939    5221    5222    5266    5267    5311    5312    5313    5314
                5392    5393    5394    5436    5432    5434    5446    5476    5477    5516    5517    5536    5541    5542    5543
                5576    5577    5578    5691    5692    5601    5618    5619    5650    5651    5695    5696    5697    5698    5699
                5709    5710    5712    5713    5722    5723    5727    5728    5738    5739    5740    5746    5747    5748    5749
                5756    5771    5782    5783    5784    5789    5790    5795    5796    5798    5811    5828    5829    5830    5831
                5841    5844    5882    5883
.ERR            720     722     724     726     728     730     732     734     736     738     740     742     744     746
                1738    1743    1746    1767    1774    1777    1835    1836    1842    1845    1852    1868    1871    1874
                1897    1914    1920    1923    1940    1945    1949    1966    1974    1977    1998    1999    2005    2008
                2031    2034    2051    2057    2060    2077    2083    2086    2103    2109    2112    2129    2135    2138
                2163    2166    2187    2192    2195    2209    2216    2219    2230    2231    2237    2240    2256    2259
                2287    2293    2296    2315    2321    2324    2343    2349    2352    2371    2377    2380    2398    2401
                2408    2425    2431    2434    2451    2457    2460    2477    2483    2486    2503    2509    2526    2529
                2538    2555    2561    2564    2581    2587    2590    2607    2614    2617    2644    2651    2654    2657
                2694    2707    2708    2714    2717    2734    2740    2743    2760    2766    2769    2786    2792    2795
                2818    2821    2838    2844    2847    2864    2870    2873    2890    2896    2899    2915    2926    2929
                2961    2971    3020    3027    3030    3063    3070    3074    3174    3187    3190    3232    3239    3242
                3299    3305    3308    3363    3380    3387    3390    3409    3423    3426    3487    3493    3496    3509    3512
                3622    3567    3576    3579    3616    3631    3634    3687    3693    3696    3831    3840    3843    3889    3892
                3901    3948    3954    3957    4066    4080    4093    4110    4124    4127    4297    4305    4308    4337    4340
                4348    4397    4413    4416    4645    4654    4657    4695    4703    4706    4799    4808    4818    4821    4827
                4886    4889    4928    4935    4938    5210    5218    5221    5255    5263    5266    5300    5308    5311    5317
                5353    5353    5385    5390    5393    5425    5430    5433    5466    5473    5476    5507    5513    5516    5522
                5541    5557    5563    5566    5582    5589    5591    5608    5615    5618    5640    5647    5650    5653
                5703    5733    5743    5746    5777    5786    5789    5823    5832    5835    5870    5879    5882    5885
.ERR            528     542     563     582     608     610     620     644     646     720     722     724     726     728
                730     732     734     736     738     740     742     744     746     748     803     837     990     1038    1485
                1738    1741    1748    1767    1772    1779    1836    1840    1846    1862    1866    1872    1889    1892    1898
                1914    1918    1924    1940    1944    1950    1966    1972    1978    1999    2003    2009    2025    2029    2032
                2051    2055    2061    2077    2081    2087    2103    2107    2113    2129    2133    2139    2155    2161    2164
                2187    2190    2196    2209    2214    2220    2231    2235    2241    2259    2263    2269    2287    2291    2294
                2315    2319    2325    2343    2347    2353    2371    2375    2381    2399    2403    2409    2425    2429    2432
                2451    2455    2461    2477    2481    2487    2503    2507    2513    2529    2533    2539    2555    2559    2562
.MACRO          1
.REGS           528
                542     563     582     608     610     620     644     646     720     722     724     726     1038
                730     732     734     736     738     740     742     744     746     748     803     837     990     1485
                1738    1741    1748    1767    1772    1779    1836    1840    1846    1862    1866    1872    1889    1898
                1914    1918    1924    1940    1944    1950    1966    1972    1978    1999    2003    2009    2025    2032
                2051    2055    2061    2077    2081    2087    2103    2107    2113    2129    2133    2139    2155    2164
                2187    2190    2196    2209    2214    2220    2231    2235    2241    2259    2263    2269    2287    2294
                2315    2319    2325    2343    2347    2353    2371    2375    2381    2399    2403    2409    2425    2432
                2451    2455    2461    2477    2481    2487    2503    2507    2513    2529    2533    2539    2555    2562
```

Spooler runtime 24 Seconds, 711 KCS, 681 disk reads, 3 disk writes, 144 pages

[redacted]