DV11

BASIC RIW + ROM INSTRUCTION

MD-11-DZDVA-B

EP DZDVA B DL A

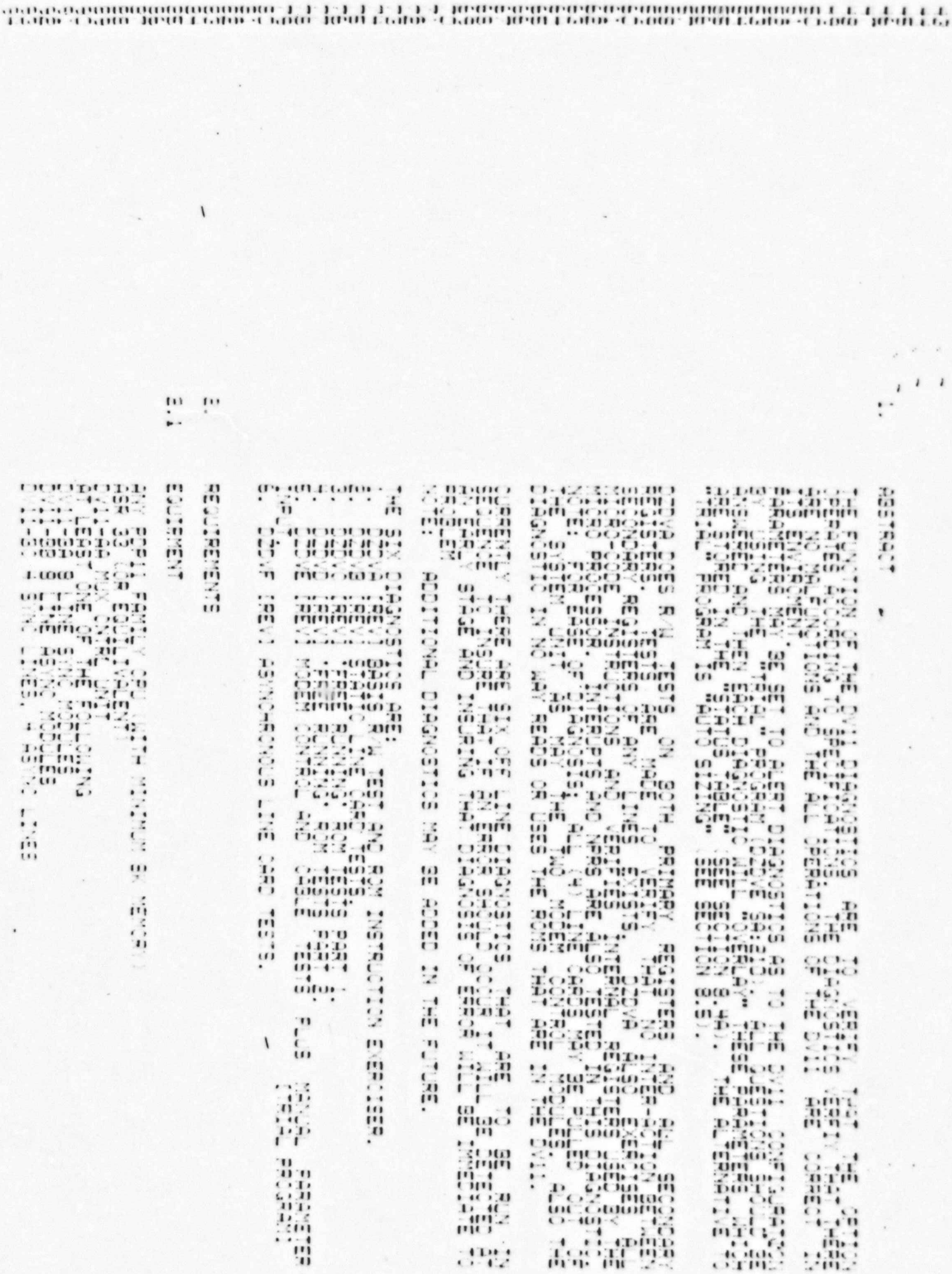
OCT 1976

FICHE 1 OF 1

digital

Made In U.S.A.

801



2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABL AND BOOTSTRAP LOADER RESIDE. LOCATION 1800 THRU 1F00 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER DVAL TRIAL PROGRAM HAS BEEN EDITED, OR AFTER THE 'AUTO SIZING' HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

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

ABSOLUTE LOADER STARTING ADDRESS =500

MEMORY + SIZE

4K	1F
8K	3F
12K	5F
16K	7F
20K	9F
24K	BF
28K	DF
32K	FF

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

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

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

4. STARTING PROCEEDURE

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

```
'MAP OF DV11 STATUS'
 1500    176000
 1502    176010
 1504    176020
 1506    176030
 1510    176040
 1512    176050
 1514    176060
 1516    176070
 1520    176100
```

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

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

4.1 CONTROL SWITCH SETTINGS

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

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

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DV11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DV11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS FOUR DV11S: 01,02,03 WILL BE SET IN LOC 'DVACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW00) ALTERS THAT LOCATION;THEREFORE IF FOUR DV11S ARE IN THE SYSTEM ***DO NOT*** SET SWITCHS GREATER THAN SW 03 IN THE UP POSITION. THIS WOULD BE A FATAL ERROR. DO NOT SELECT MORE ACTIVE DV11S THAN HAS BEEN GIVEN INFORMATION ABOUT IN TRIAL PROGRAM.

METHOD: A: LOAD ADDRESS 200
B: START WITH SW 00=1
C: PROGRAM WILL TYPE MESSAGE
D: SET THE BINARY NUMBER OF DV11S DESIRED ACTIVE EXAMPLE: 1=1 DV11; 3=2 DV11; 7=3 DV11; 17=4 DV11 37=5 DV11 ETC. PRESS CONTINUE.
E: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/03)
F: SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST. IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HASS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE THE REASON BEING IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS. ALSO WHEN A TEST IS SELECTED ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA; THIS SWITCH WILL ONLY WORK IF CALL 'SCOPE' IS IN THAT TEST. THE REASON BEING THAT MOST TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE THUS IN BLOCK DATA; THE PATTERN CANNOT BE SINGLED OUT.

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 08 GOTO BEGINNING OF THE TEST(ON ERROR).
5. SW 10 GOTO NEXT TEST(ON ERROR).

SCOPE SWITCHES

1. SW 09 (IF ENABLED BY 'SCOPE') ON AN ERROR; IF AN '*' IS PRINTED IN FRONT OF THE TEST NO. (EX. *TEST NO. 10) SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS *USUALLY* THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0). IF SW09 IS NOT ENABELED: AND THERE IS A *HARD* ERROR (CONSTANT): SW08 IS BEST.
    (SW14=1,0, SW10=0, SW09=0, SW08=1). FOR INTERMITTENT ERRORS: SW14=1 WILL LOOP ON TEST REGUARDLESS OF ERROR OR NOT ERROR.
    (SW14=1, SW10=0, SW09=0, SW08=1.0)
2. SW 14
3. SW 11

4.2 STARTING ADDRESS

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

NOTE: IF ADDRESS 000048 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY AFTER ALL AVAILABLE DV11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5. OPERATING PROCEDURE

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

AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

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

6. ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.2 ERROR RECOVERY

IF FOR SOME REASON THE DV11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN; LOOK IN LOCATION 'TSTNO' (ADDRESS 1224)FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DV11 WAS DOING AT THE TIME OF THE ERROR.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)
STATUS TABLE SHOULD BE VERIFIED REGUARDLESS OF HOW PROGRAM WAS STARTED.
ALSO IT IS IMPORTANT TO USE THIS LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETLY ISOLATE PROBLEMS.

7.2 OPERATING RESTRICTIONS

DV11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE FIRST AND ONLY THE FIRST RUNNING OF ANY DV11 DIAGNOSTIC IF "AUTO SIZING" IS NOT USED.
NOTE: IF NO PROGRAM OTHER THAN A DV11 DIAGNOSTIC WAS LOADED AFTER DV11 TRIAL OR IF CORE MEMORY HAS NOT BEEN CHANGED;OR IF THERE IS NO DV11 CONFIGURATION CHANGES; THE DV11 TRIAL PROGRAM NEED NEVER BE RUN AGAIN. HOWEVER IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DV11 TRIAL PROGRAM MUST BE RUN AGAIN BEFORE RUNNING THE DIAGNOSTICS NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING THE 'AUTO SIZING' WHEN PROGRAM IS INITIALLY STARTED WITH SW07=0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS (SYNC LINE CARDS ONLY)

1. HARDWARE MUST BE SET TO FULL DUPLEX
2. PARITY OFF.
3. ALL LINES OF A PARTICULAR LINE CARD MUST BE CONFIGURED THE SAME.

8. MISCELLANEOUS

8.1 EXECUTION TIME

ALL DV11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 4 MINS. THIS IS ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED: THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO 'VERIFY NO *HARD* ERRORS' AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS -EACH TIME PROGRAM IS STARTED- WILL BE A 'QUICK PASS' UNTILL ALL DV11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS 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 LENGHTLY. THEREFORE EACH CHAR IS AN "END PASS AND THE ENTIRE "END" IS NOT REQUIRED FOR ACCEPTANCE.

8.4 KEY LOCATIONS

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

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

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

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.

DVACTV (1276) EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DV11 WILL BE TESTED IN TURN. EXAMPLE: (DVACTV) 1276/0000000000011111 MEANS THAT DV11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DVACTV) 1276/0000000000010001 MEANS THAT DV11 NO. 00,04 WILL BE TESTED.

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

L00.03 (1412)
L04.07 (1414)
L08.11 (1416)
L12.15 (1420) CONTAINS THE STATUS OF THE CURRENT DV11 UNDER TEST.

BIT 15	SET:	LINE CARD *NOT INSTALLED (AND WONT BE TESTED)
BIT 14	SET:	RESERVED
BIT 13	SET:	RESERVED
BIT 12	SET:	ONE SYNC, =0: TWO SYNCS.
BIT 11	SET:	ASYNC LINE CARD, =0 SYNC LINE CARD.
BIT 10	SET:	RESERVED
BIT 09	SET:	BITS PER CHAR. (USED WITH BIT8)
BIT 08	SET:	BITS PER CHAR. (USED WITH BIT9)

BIT09	BIT08	BITS PER CHAR.
0	0	8
0	1	7
1	0	6
1	1	5

BIT 07-00 SYNC "A" FOR SPECIFIED LINE CARD. BITS 07-00 MUST BE ALL ZEROS FOR TESTING 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 HANG (TOGGLE IN) IS DESIRED. IN THIS WAY 95% OF THE PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.
THEREFORE:

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

```
                                ;*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
```

```
PDP-11 MACRO11 47(730) 17-SEP-76  14:02  PAGE 15
DEVAB.P11       GENERAL DEFINATIONS AND EQUIVALENCIES


                                ;REGISTER DEFINITIONS
                                ;--------------------

000000                          R0=%0           ;GENERAL REGISTER
000001                          R1=%1           ;GENERAL REGISTER
000002                          R2=%2           ;GENERAL REGISTER
000003                          R3=%3           ;GENERAL REGISTER
000004                          R4=%4           ;GENERAL REGISTER
000005                          R5=%5           ;GENERAL REGISTER
000006                          SP=%6           ;PROCESSOR STACK POINTER
000007                          PC=%7           ;PROGRAM COUNTER

                                ;LOCATION EQUIVALENCIES
                                ;----------------------

177776                          PS=177776       ;PROCESSOR STATUS WORD
001200                          STACK=1200      ;START OF PROCESSOR STACK

000020                          BIT4=000020
000040                          BIT5=000040
000100                          BIT6=000100
000200                          BIT7=000200
000400                          BIT8=000400
001000                          BIT9=001000
002000                          BIT10=002000
004000                          BIT11=004000
010000                          BIT12=010000
020000                          BIT13=020000
040000                          BIT14=040000
100000                          BIT15=100000

                                ;-----------------------------------
                                PRIO=BIT10
                                RPACT=BIT10
                                XMIM=BIT10
                                XFER=BIT10+BIT12
                                NXFR=BIT10+BIT11
                                CLOC=BIT10+BIT11+BIT12
                                BUSY=BIT10+BIT11
                                ERR=BIT10+BIT11+BIT12
                                ;-----------------------------------
```

```
                                        ;;********************************************************************
                                        ;----------------------------------------------------------------------
                                        ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
                                        ;THE STANDARD "TRAP CATCHER" IS PLACED
                                        ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                        ;IT LOOKS LIKE "PC+2 HALT".
                                        ;----------------------------------------------------------------------
                                        ;;********************************************************************

                  000000                .=0
                                        ;STANDARD INTERRUPT VECTORS
                                        ;-------------------------

                  000024                .=24
000024  000000'                         .PFAIL                  ;POWER FAIL HANDLER
000026  000340                         340                     ;SERVICE AT LEVEL 7
000030  000000'                         .ERR                    ;ERROR HANDLER
000032  000340                         340                     ;SERVICE AT LEVEL 7
000034  000000'                         .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                         340                     ;SERVICE AT LEVEL 7
                  000040                .=40
000040  000000                         .BLKW 1                 ;SAVE FOR ACT-11 OR DDT2
000042  000000                         .BLKW 1                 ;RETURN ADDRESS IF UNDER ACT-11 OR DDT2
000044  000000                         .BLKW 1                 ;SAVE FOR ACT-11 OR DDT2
000046  000000                         LOGICAL                 ;FOR USE WITH ACT-11 OR DDT2

                  000174                .=174
000174  000000                 LIGHT:  0
                  000176                .=176
000176  000000                 SWR:    0

                  000200                .=200
000200  000137  001742                 JMP     .START          ;GO TO START OF PROGRAM

                  001000                .=1000
001000  036077  040616  047111 MTITLE: .ASCIZ  <377><12><MAINDEC-11-DCDVA-B><377>BASIC DV11 CONTROLLER MODULES TESTING

                  001200                .=1200
001200                          LIGHTS:
001200  177570                         177570
001202  177570                 SWR:    177570
                                        ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                                        ;-------------------------------------------------

001204  177560                 TKSR:   177560                  ;TELETYPE KEYBOARD CONTROL REGISTER
001206  177562                 TKDBR:  177562                  ;TELETYPE KEYBOARD DATA BUFFER
001210  177564                 TPSR:   177564                  ;TELEPRINTER CONTROL REGISTER
001212  177566                 TPDBR:  177566                  ;TELEPRINTER DATA BUFFER

                                        ;PROGRAM CONTROL PARAMETERS
                                        ;-------------------------

001214  000000                 RETURN: 0                       ;SCOPE ADDRESS FOR LOOP ON TEST
001216  000000                 NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001220  000000                 LOOP:   0                       ;ADDRESS FOR LOOP ON CURRENT DATA
```

```
DSRVAB-B MACY11 27(738)  12-SEP-78  14:08  PAGE 17
DSRVAB.P11      PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

                                        ICOUNT: 3               ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
                                        LCNT:   0               ;NUMBER OF ITERATIONS COMPLETED
                                        TSTNO:  0               ;NUMBER OF TEST IN PROGRESS
                                        PASSCNT: 0              ;NUMBER OF PASSES COMPLETED
                                        ERRCNT: 0               ;TOTAL NUMBER OF ERRORS
                                        LSTERR: 0               ;PC OF LAST ERROR CALL

                                                ;PROGRAM VARIABLES
                                                ;-----------------

                                        STAT:   0               ;DV STATUS WORD STORAGE
                                        SYNCX:  .BLKW 1
                                        MPRGX:  .BLKW 1
                                        TEMP:   .BLKW 1         ;TEMPORARY STORAGE
                                        TEMP1:  .BLKW 1         ;TEMPORARY STORAGE
                                        TEMP2:  .BLKW 1         ;TEMPORARY STORAGE
                                        TEMP3:  .BLKW 1         ;TEMPORARY STORAGE
                                        ERRPC:  .BLKW 1         ;ERROR STORAGE
                                        ERRPC1: .BLKW 1         ;ERROR STORAGE
                                        ERRPC2: .BLKW 1         ;ERROR STORAGE
                                        ERRPC3: .BLKW 1         ;ERROR STORAGE
                                        ERRPC4: .BLKW 1         ;ERROR STORAGE
                                        ERRPC5: .BLKW 1         ;ERROR STORAGE
                                        SVPC:   .BLKW 1         ;STACK POINTER STORAGE
                                        SVR5:   .BLKW 1         ;PROGRAM COUNTER STORAGE
                                        SVR4:   .BLKW 1         ;STATUS STORAGE
                                        DVNUM:  .BLKW 1         ;TOTAL NUMBER OF DVI:S.
                                        SAVACT: .BLKW 1         ;ORIGINAL ACTV. DEVICES.
                                        SAVNUM: .BLKW 1         ;WORKABLE NUMBER.
                                        RUN:    .BLKW 1         ;POINTER ONE PAST RUNNING DEVICE.
                                        .EVEN
                                        DREAM:  DV.MAP          ;TABLE POINTER.
```

```
RSDVA-B MACY11 27(732)  17-SEP-76  14:02  PAGE 18
RSDVAB.P11      PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.


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

001310  000000          INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001311  000000          ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
001312  000000          LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
001313  000000          QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG.
                                                        ;ON FIRST PASS OF EACH DV11 ITERATIONS WILL BE SUPPRESSE

                        .EVEN
                000000  $Y=0

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

                        ;;**************************************************************
                        ;--------------------------------------------------------------
001314                  TRPTAB:
001314  104400          SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001314  002634                  SCOPE
001316  104401          SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001316  000000                  SCOP1
001320  104402          TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001320  002744                  TYPE
001322  104403          INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001322  003104                  INSTR
001324  104404          INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001324  003244                  INSTER
001326  104405          PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001326  003244                  PARAM
001330  104406          SAVOS=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
001330  003444                  SAVOS
001332  104407          RESOS=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
001332  003504                  RESOS
001334  104410          CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
001334  003536                  CONVRT
001336  104411          CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR LF.
001336  003540                  CNVRT
001340  104412          MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
001340  004600                  MSTCLR
001342  104413          RAMCLR=TRAP+13          ;CALL TO CLEAR THE RAMS
001342  004616                  RAMCLR
001344  104414          DELAY=TRAP+14           ;CALL TO VARIABLE DELAY COUNTER
001344  004476                  DELAY
001346  104415          ROMCLK=TRAP+15          ;CALL TO CLOCK ROM ONCE
001346  004660                  ROMCLK
001350  104416          DATACLK=TRAP+16         ;CALL TO CLK DATA
001350  004576                  DATACLK

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

F02

```
DZDVA-B MACY11 27(732)  17-SEP-76  14:02  PAGE 19
DZDVAB.P11      PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

                                        ;DV11 VECTOR AND REGISTER INDIRECT POINTERS
001360  000000          DVRVEC: 0               ;POINTER TO DV11 RECEIVER INTERRUPT VECTOR
001362  000000          DVRLVL: 0               ;POINTER TO DV11 RECEIVER INTERRUPT SERVICE PS
001364  000000          DVTVEC: 0               ;POINTER TO DV11 TRANSMITTER INTERRUPT VECTOR
001366  000000          DVTLVL: 0               ;POINTER TO DV11 TRANSMITTER INTERRUPT SERVICE PS
001370  000000          DVSCR:  0               ;POINTER TO DV11 SYSTEM CONTROL REGISTER
001372  000000          DVSCRH: 0               ;POINTER TO DV11 SYSTEM CONTROL REGISTER HIGH BYTE.
001374  000000          DVRCC:  0               ;POINTER TO DV11 NEXT RECEIVED CHARACTER REGISTER
001376  000000          DVLCR:  0               ;POINTER TO DV11 LINE PRAMETER REGISTER
001400  000000          DVSRS:  0               ;POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER
001402  000000          DVSRSH: 0               ;POINTER TO DV11 SECONDARY REGISTER SELECT HIGH BYTE.
001404  000000          DVSRA:  0               ;POINTER TO DV11 SECONDARY REGISTER ACCESS REGISTER
001406  000000          DVSFR:  0               ;POINTER TO DV11 SPECIAL FUNCTIONS REGISTER
001410  000000          DVNSR:  0               ;POINTER TO DV11 NPR STATUS REGISTER
001412  000000          RESV16: 0               ;POINTER TO RESERVED REGISTER.

                                        ;DV11 CONTROL INDICATORS FOR CURRENT DV11 UNDER TEST
                                        ;---------------------------------------------------
001414  000             MASK.A: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 00-03
001415  000             MASK.B: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 04-07
001416  000             MASK.C: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 08-11
001417  000             MASK.D: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 12-15

001420  010             CLK.A:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03
001421  010             CLK.B:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07
001422  010             CLK.C:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11
001423  010             CLK.D:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 12-15

001424  000000          L00.03: 000000          ;PARAMETERS FOR LINES 00-03
001426  000000          L04.07: 000000          ;PARAMETERS FOR LINES 04-07
001430  000000          L08.11: 000000          ;PARAMETERS FOR LINES 08-11
001432  000000          L12.15: 000000          ;PARAMETERS FOR LINES 12-15

001434  000000          SYNC2A: 000000          ;SYNC 2
001436  000000          SYNC2B: 000000          ;
001440  000000          SYNC2C: 000000          ;
001442  000000          SYNC2D: 000000          ;

                                        ;SUMMARY
                                        ;-------
                                ;       MASK.X          040     5 BITS PER CHAR.
                                ;                       100     6 BITS PER CHAR.
                                ;                       200     7 BITS PER CHAR.
                                ;                       000     8 BITS PER CHAR.

                                ;       CLK.X           005     5 BITS PER CHAR.
                                ;                       006     6 BITS PER CHAR.
                                ;                       007     7 BITS PER CHAR.
                                ;                       010     8 BITS PER CHAR.
```

```
DECVAB-B MACY11 27(732) 17-SEP-76 14:08  PAGE 20
DECVAB.P11      PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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

                001500                  .=1500
                                DV.MAP:
001500  000001         DVCR00: .BLKW 1        ;CONTROL STATUS REGISTER FOR DV11 NUMBER 00
001502  000001         DVTR00: .BLKW 1        ;VECTOR "A" FOR DV11 NUMBER 00
001504  000001         DVO0.A: .BLKW 1        ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
001506  000001         SYNA00: .BLKW 1        ;SYNC TWO
001510  000001         DVO0.B: .BLKW 1        ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
001512  000001         SYNB00: .BLKW 1        ;SYNC TWO
001514  000001         DVO0.C: .BLKW 1        ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
001516  000001         SYNC00: .BLKW 1        ;SYNC TWO
001520  000001         DVO0.D: .BLKW 1        ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
001522  000001         SYND00: .BLKW 1        ;SYNC TWO

001524  000001         DVCR01: .BLKW 1        ;CONTROL STATUS REGISTER FOR DV11 NUMBER 01
001526  000001         DVTR01: .BLKW 1        ;VECTOR "A" FOR DV11 NUMBER 01
001530  000001         DV01.A: .BLKW 1        ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
001532  000001         SYNA01: .BLKW 1        ;SYNC TWO
001534  000001         DV01.B: .BLKW 1        ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
001536  000001         SYNB01: .BLKW 1        ;SYNC TWO
001540  000001         DV01.C: .BLKW 1        ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
001542  000001         SYNC01: .BLKW 1        ;SYNC TWO
001544  000001         DV01.D: .BLKW 1        ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
001546  000001         SYND01: .BLKW 1        ;SYNC TWO

001550  000001         DVCR02: .BLKW 1        ;CONTROL STATUS REGISTER FOR DV11 NUMBER 02
001552  000001         DVTR02: .BLKW 1        ;VECTOR "A" FOR DV11 NUMBER 02
001554  000001         DV02.A: .BLKW 1        ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
001556  000001         SYNA02: .BLKW 1        ;SYNC TWO
001560  000001         DV02.B: .BLKW 1        ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
001562  000001         SYNB02: .BLKW 1        ;SYNC TWO
001564  000001         DV02.C: .BLKW 1        ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
001566  000001         SYNC02: .BLKW 1        ;SYNC TWO
001570  000001         DV02.D: .BLKW 1        ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
001572  000001         SYND02: .BLKW 1        ;SYNC TWO

001574  000001         DVCR03: .BLKW 1        ;CONTROL STATUS REGISTER FOR DV11 NUMBER 03
001576  000001         DVTR03: .BLKW 1        ;VECTOR "A" FOR DV11 NUMBER 03
001600  000001         DV03.A: .BLKW 1        ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
001602  000001         SYNA03: .BLKW 1        ;SYNC TWO
001604  000001         DV03.B: .BLKW 1        ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
001606  000001         SYNB03: .BLKW 1        ;SYNC TWO
001610  000001         DV03.C: .BLKW 1        ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
001612  000001         SYNC03: .BLKW 1        ;SYNC TWO
001614  000001         DV03.D: .BLKW 1        ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 03
001616  000001         SYND03: .BLKW 1        ;SYNC TWO

001620  000001         DVCR04: .BLKW 1        ;CONTROL STATUS REGISTER FOR DV11 NUMBER 04
001622  000001         DVTR04: .BLKW 1        ;VECTOR "A" FOR DV11 NUMBER 04
001624  000001         DV04.A: .BLKW 1        ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
001626  000001         SYNA04: .BLKW 1        ;SYNC TWO
001630  000001         DV04.B: .BLKW 1        ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
001632  000001         SYNB04: .BLKW 1        ;SYNC TWO
001634  000001         DV04.C: .BLKW 1        ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04
```

```
001636  000001          SYNC04: .BLKW 1         ;SYNC TWO
001640  000001          DV04.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
001642  000001          SYND04: .BLKW 1         ;SYNC TWO

001644  000001          DVCR05: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 05
001646  000001          DVTR05: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 05
001650  000001          DV05.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
001652  000001          SYNA05: .BLKW 1         ;SYNC TWO
001654  000001          DV05.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
001656  000001          SYNB05: .BLKW 1         ;SYNC TWO
001660  000001          DV05.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
001662  000001          SYNC05: .BLKW 1         ;SYNC TWO
001664  000001          DV05.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
001666  000001          SYND05: .BLKW 1         ;SYNC TWO

001670  000001          DVCR06: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 06
001672  000001          DVTR06: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 06
001674  000001          DV06.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
001676  000001          SYNA06: .BLKW 1         ;SYNC TWO
001700  000001          DV06.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
001702  000001          SYNB06: .BLKW 1         ;SYNC TWO
001704  000001          DV06.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
001706  000001          SYNC06: .BLKW 1         ;SYNC TWO
001710  000001          DV06.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
001712  000001          SYND06: .BLKW 1         ;SYNC TWO

001714  000001          DVCR07: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 07
001716  000001          DVTR07: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 07
001720  000001          DV07.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
001722  000001          SYNA07: .BLKW 1         ;SYNC TWO
001724  000001          DV07.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
001726  000001          SYNB07: .BLKW 1         ;SYNC TWO
001730  000001          DV07.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
001732  000001          SYNC07: .BLKW 1         ;SYNC TWO
001734  000001          DV07.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 07
001736  000001          SYND07: .BLKW 1         ;SYNC TWO

001740  000000          DV.END: 000000
```

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

001740  012737  000340  177776  .START: MOV     #340,@#PS               ;LOCK OUT INTERRUPTS
001746  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
001752  012737  004402  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
001760  113737  001301  001303          MOVB    DVNUM,SAVNUM            ;SAVE NUMBER OF DEVICES IN SYSTEM.
001770  005037  001230                  CLR     PASCNT                  ;CLEAR PASS COUNT
001774  105037  001311                  CLRB    ERRFLG                  ;CLEAR ERROR FLAG
002000  105037  001313                  CLRB    QV.FLG                  ;ZERO QUICK VERIFY FLAG
002004  012737  001500  001306          MOV     #DV.MAP,CREAM           ;GET MAP POINTER.
002012  112737  000001  001304          MOVB    #1,RUN                  ;POINT POINTER TO FIRST DEVICE.
002020  005037  001232                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
002024  005037  001234                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
002030  012737  000001  001226          MOV     #1,TSTNO                ;SET UP FOR TEST 1
002036  012737  001742  001314          MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
002044  105737  001310                  TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
002050  001063                          BNE     1$                      ;BR IF YES
002052  013746  000004                  MOV     4,-(SP)                 ;
002056  013746  000006                  MOV     6,-(SP)                 ;
002062  005037  000006                  CLR     6                       ;
002066  012737  002104  000004          MOV     #80$,4                  ;
002074  005777  177102                  TST     @SWR                    ;
002100  000240                          NOP                             ;
002102  000407                          BR      81$                     ;
002104  022626                  80$:    CMP     (SP)+,(SP)+             ;
002106  012737  000174  001600          MOV     #LIGHT,LIGHTS           ;
002114  012737  000176  001602          MOV     #@SWR,SWR               ;
002122  012637  000006          81$:    MOV     (SP)+,6                 ;
002126  012637  000004                  MOV     (SP)+,4                 ;
002132  104402  001000                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
002136  105137  001310                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
002142  105777  177034                  TSTB    @SWR                    ;BIT7=1??
002146  100402                          BMI     16$                     ;BR IF NO AUTO SIZE
002150  004737  006624                  JSR     PC,OSRMAP               ;GO DO THE AUTO SIZE
002154  104402  005461          16$:    TYPE    ,XHEAD          ;TYPE HEADER
002160  012737  001500  001246          MOV     #DV.MAP,TEMP1           ;SET POINTER
002166  017737  177054  001250  5$:     MOV     @TEMP1,TEMP2            ;SET DATA
002174  022737  177777  001250          CMP     #177777,TEMP2           ;ALL DONE?
002202  001406                          BEQ     1$                      ;BR IF YES
002204  104410                          CONVRT                          ;
002206  005506                          XSTATQ                          ;
002210  062737  000002  001246          ADD     #2,TEMP1                ;UPDATE POINTER
002216  000763                          BR      5$
002220  005737  000042          1$:     TST     @#42                    ;IS PROGRAM RUNNING UNDER MONITOR
002224  001030                          BNE     3$                      ;BR IF YES
002226  032777  000001  176746          BIT     #SW00,@SWR              ;SELECT SPECIFIC DEVICES``
002234  001424                          BEQ     3$                      ;BR IF NO.
002236  104402  005402                  TYPE    ,MNEW                   ;TYPE THE MESSAGE.
002242  005000                          CLR     R0                      ;ZERO DATA LIGHTS
```

```
                                                HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
002244  000000                                  CMPB    @SWR,SAVACT     ;IS THE NUMBER VALID?
002246  127737  176730  001302                BLOS    2$              ;BR IF NUMBER IS OK.
002254  101404                                  TYPE    .MERR3          ;TELL USER OF INVALID NUMBER.
002256  104402  005243                          HALT                    ;STOP EVERY THING.
002262  000000                                  BR      .-2             ;RESTART THE PROGRAM AGAIN.
002264  000776                          2$:     MOVB    @SWR,DVACTV     ;GET NEW DEVICE PATTERN
002266  117737  176710  001300                MOVB    DVACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
002274  113700  001300                          BIC     #^C<377>,R0     ;USE ONLY LOW BYTE.
002300  042700  177400                          HALT                    ;CONTINUE DYNAMIC SWITCHES.
002304  000000                          3$:     MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
002306  012700  000300                          MOV     #302,R1         ;VECTOR AREA. 300-776
002312  012701  000302                  4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
002316  010120                                  CLR     (R1)+           ;IN VECTOR AREA.
002320  005021                                  CMP     (R0)+,(R1)+     ;POP POINTERS
002322  022021                                  CMP     #1000,R0        ;ALL DONE??
002324  022700  001000                          BNE     4$              ;BR IF NO.
002330  001372

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

002332  012737  000340  177776  .BEGIN: MOV     #340,PS         ;LOCK OUT INTERRUPTS
002340  012706  001200                  MOV     #STACK,SP       ;SET UP STACK
002344  005737  000042                  TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
002350  001023                          BNE     3$              ;BR IF YES
002352  032777  000004  176622          BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
002360  001411                          BEQ     1$              ;BR IF NO LOCK DESIRED.
002362  104402  005301                  TYPE    .MLOCK          ;TYPE LOCK SELECTED.
002366  012737  000240  002702          MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
002374  012737  000240  002704          MOV     #NOP,TTST+2     ;SET UP TO LOCK
002402  000406                          BR      2$              ;CONTINUE ALONG.
002404  013737  003014  002702  1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
002412  013737  003016  002704          MOV     BRX,TTST+2      ;LOCK NOT SELECTED. SET UP FOR NORMAL SCOPE LOOP
002420                                  2$:
002420  012737  005666  001214  3$:     MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
002426  104402  005171          4$:     TYPE    ,MR             ;TYPE R
002432  000177  176666                  JMP     @RETURN         ;START TESTING
```

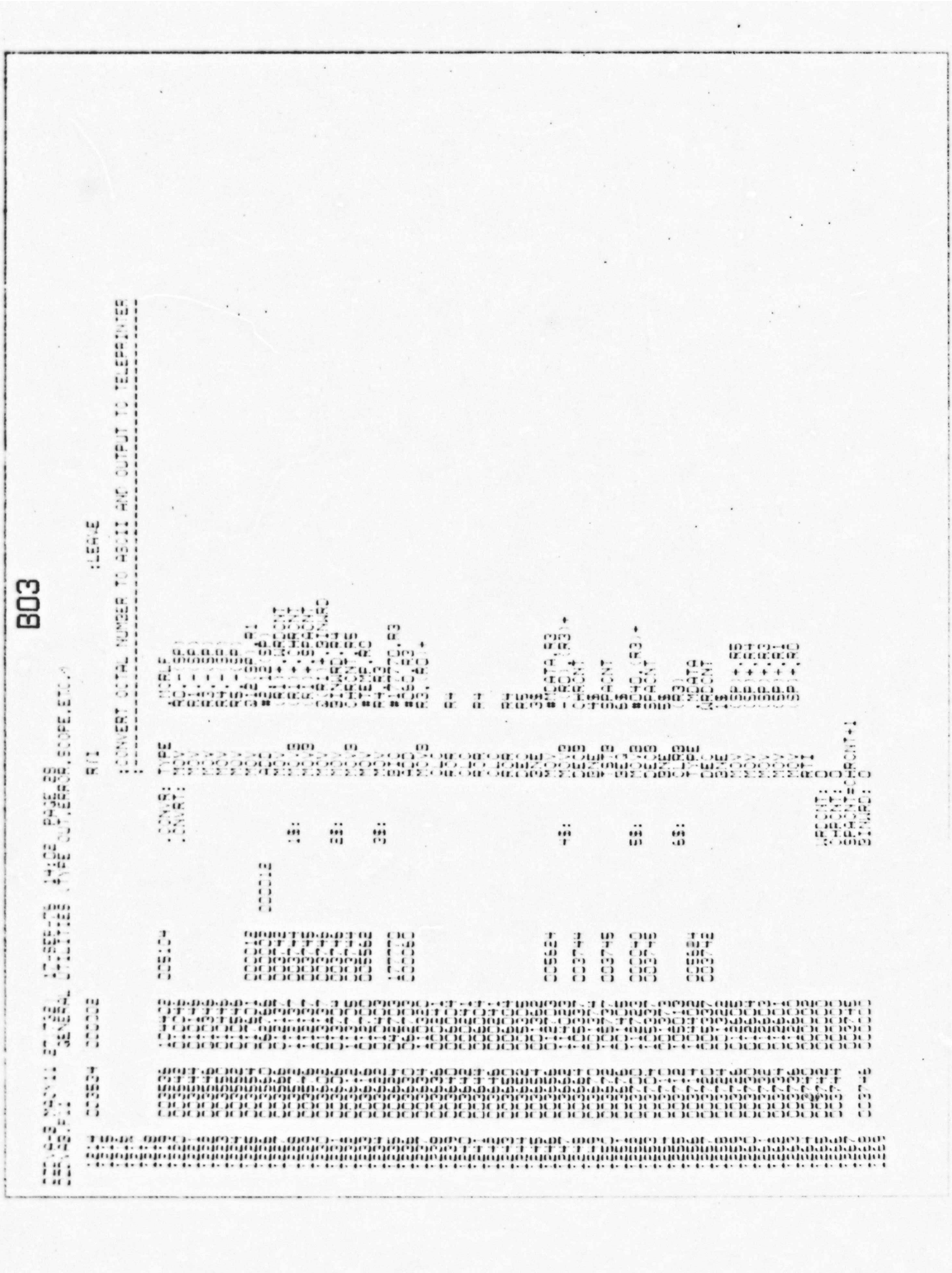
```
                                                        ;END OF PASS
                                                        ;TYPE NAME OF TEST
                                                        ;UPDATE PASS COUNT
                                                        ;CHECK FOR EXIT TO ACT-11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

```
1046  002654  122726  000007                      CMPB    #7,(SP)+        ;WAS IT CNTRL 'G' ?
1047  002660  001002                              BNE     .+6             ;BR IF NO.
1048  002662  004737  004640                      JSR     PC,SERV.G       ;SERVICE "CNTRL 'G'".
1049  002666  005037  001234          64$:        CLR     LSTERR          ;CLEAR LAST ERROR PC.
1050  002672  010016                              MOV     R0,(SP)         ;SAVE R0 ON THE STACK
1051  002674  032777  040000  176300              BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
1052  002702  001407                  TTST:       BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1; THIS LOC =240)
1053  002704  000437                              BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
1054  002706  105777  176272                      TSTB    @TKCSR          ;KEYBOAD DONE?
1055  002712  100034                              BPL     3$              ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
1056  002714  017700  176266                      MOV     @TKDBR,R0       ;CLEAR DONE BIT
1057  002720  000415                              BR      2$              ;CONTINUE
1058  002722  032777  004000  176252  1$:         BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
1059  002730  001011                              BNE     2$              ;BR IF YES
1060  002732  105737  001313                      TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
1061  002736  001406                              BEQ     2$              ;BR IF QUICK PASS.
1062  002740  005237  001224                      INC     LPCNT           ;UPDATE ITERATION COUNTER
1063  002744  023737  001224  001222              CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
1064  002752  001014                              BNE     3$              ;BR IF NOT YET
1065  002754  105037  001311          2$:         CLRB    ERRFLG          ;PREPARE FOR NEW TEST
1066  002760  005037  001224                      CLR     LPCNT           ;START ICOUNTER AT 0
1067  002764  005037  001220                      CLR     LOCK
1068  002770  012737  000020  001222              MOV     #20,ICOUNT      ;RESET ITERATIONS
1069  002776  013737  001216  001214              MOV     NEXT,RETURN     ;GET NEXT TEST
1070  003004  011600                  3$:         MOV     (SP),R0         ;POP R0 OFF OF THE STACK
1071  003006  022626                              POP2SP                  ;FAKE AN "RTI"
1072  003010  000177  176200                      JMP     @RETURN         ;GO DO THE TEST
1073  003014  001407                  BRW:        1407
1074  003016  000437                  BRX:        437
1075
1076                                              ;CHECK FOR FREEZE ON CURRENT DATA
1077                                              ;-------------------------------
1078
1079  003020  032777  001000  176154  .SCOP1:     BIT     #SW09,@SWR      ;IS SW09=1(SET)?
1080  003026  001405                              BEQ     1$              ;BR IF NOT SET.
1081  003030  005737  001220                      TST     LOCK
1082  003034  001402                              BEQ     1$
1083  003036  013716  001220                      MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
1084  003042  000002                  1$:         RTI                     ;GO BACK.
1085
1086                                              ;TELETYPE OUTPUT ROUTINE
1087                                              ;----------------------
1088
1089  003044  010546                  .TYPE:      MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
1090  003046  017605  000002                      MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
1091  003052  062766  000002  000002              ADD     #2,2(SP)        ;POP OVER ADDRESS.
1092  003060  032777  010000  176114  1$:         BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
1093  003066  001012                              BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
1094  003070  105715                              TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
1095  003072  100002                              BPL     2$              ;BR IF NUMBER IS PLUS
1096  003074  104402  005104                      TYPE    ,MCRLF          ;TYPE A CR/LF!
1097  003100  105777  176104          2$:         TSTB    @TPCSR          ;TTY READY?
1098  003104  100375                              BPL     2$              ;BR IF NO.
1099  003106  112577  176100                      MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
1100  003112  001362                              BNE     1$              ;IF NOT ZERO KEEP PRINTING!
1101  003114  012605                  3$:         MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
```

```
1102  003116  000002                          RTI                     ;GO HOME
1103                                          ;-----------------------
1104
1105  003120  010346                .INSTR:   MOV     R3,-(SP)        ;SAVE R3 ON STACK
1106  003122  010446                          MOV     R4,-(SP)        ;SAVE R4 ON STACK
1107  003124  017637  000004  003142          MOV     @4(SP),.MSG
1108  003132  062766  000002  000004          ADD     #2,4(SP)
1109  003140  104402                .INST1:   TYPE
1110  003142  000000                .MSG:     0
1111  003144  012704  005520                  MOV     #INBUF,R4
1112  003150  012703  000007                  MOV     #7,R3
1113  003154  105777  176024        1$:       TSTB    @TKCSR
1114  003160  100375                          BPL     1$
1115  003162  117714  176020                  MOVB    @TKDBR,(R4)
1116  003166  142714  000200                  BICB    #200,(R4)
1117  003172  122427  000015                  CMPB    (R4)+,#15
1118  003176  001417                          BEQ     INSTR2
1119  003200  105777  176004        2$:       TSTB    @TPCSR
1120  003204  100375                          BPL     2$
1121  003206  017777  175774  175776          MOV     @TKDBR,@TPDBR
1122  003214  005303                          DEC     R3
1123  003216  001356                          BNE     1$
1124  003220  012604                          MOV     (SP)+,R4
1125  003222  012603                          MOV     (SP)+,R3
1126  003224  104402  005100        .INSTE:   TYPE    ,MQM
1127  003230  010346                          MOV     R3,-(SP)
1128  003232  010446                          MOV     R4,-(SP)
1129  003234  000741                          BR      .INST1
1130  003236  012604                INSTR2:   MOV     (SP)+,R4        ;RESTORE R4
1131  003240  012603                          MOV     (SP)+,R3        ;RESTORE R3
1132  003242  000002                          RTI
1133
1134                                          ;CONVERT ASCII STRING TO OCTAL
1135                                          ;----------------------------
1136
1137  003244  010546                .PARAM:   MOV     R5,-(SP)
1138  003246  010446                          MOV     R4,-(SP)
1139  003250  016605  000004                  MOV     4(SP),R5
1140  003254  012537  003434                  MOV     (R5)+,LOLIM
1141  003260  012537  003436                  MOV     (R5)+,HILIM
1142  003264  012537  003440                  MOV     (R5)+,DEVADR
1143  003270  112537  003442                  MOVB    (R5)+,LOBITS
1144  003274  112537  003443                  MOVB    (R5)+,ADRCNT
1145  003300  010566  000004                  MOV     R5,4(SP)
1146  003304  005005                PARAM1:   CLR     R5
1147  003306  012704  005520                  MOV     #INBUF,R4
1148  003312  122714  000015                  CMPB    #15,(R4)
1149  003316  001420                          BEQ     PARERR
1150  003320  121427  000060        1$:       CMPB    (R4),#60
1151  003324  002415                          BLT     PARERR
1152  003326  121427  000067                  CMPB    (R4),#67
1153  003332  003012                          BGT     PARERR
1154  003334  142714  000060                  BICB    #60,(R4)
1155  003340  152405                          BISB    (R4)+,R5
1156  003342  122714  000015                  CMPB    #15,(R4)
1157  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          .RESOS: 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
```

```
B03
                                                    !ERASE
                                                    ER TO ASCII AND OUTPUT TO TELEPRINTER
```



C03

```
                                                        MOV     I$,R0
                                                        .TYPE   .MASTER
                                                        .TYPE   .MTSTN
                                                        .TYPE   .XTSTN          ;SHOW XT
                                                        .TYPE   .MERRPC         ;TYPE PC.
                                                        .TYPE   .ERTAB0         ;SHOW ET
                                                        .TYPE   .MCRLF          ;GIVE A CR/LF
001311                                                  MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
                                                        TST     ERRMSG          ;IS THERE AN ERROR MESSAGE
                                                        BEQ     WRK0.FM         ;BR IF NO.
                                                        .TYPE                   ;TYPE
                                                                                ;   ERROR MESSAGE
                                                WRK0.FM:
                                                        TST     DATAHD          ;DATA HEADER?
                                                        BPO     TYPDAT          ;BR IF NO
                                                        .TYPE                   ;TYPE
                                                                                ;   DATA HEADER
                                                TYPDAT: TST     DATABP          ;DATA TABLE?
                                                        BEQ     REGRES          ;BR IF NO.
                                                        CONVRT                  ;SHOW
                                                                                ;   DATA TABLE
                                                DATABP: MOV     R0,-(SP)
                                                REGRES: RESTOR                  ;RESTORE PROC REGISTERS
                                                HALTSW: TST     #SWR            ;HALT ON ERROR?
                                                        BPL     EXITER          ;BR IF NO HALT ON ERROR
                                                        PUSHRO                  ;SAVE R0
                                                        MOV     @(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
                                                        HALT                    ;HALT
                                                        POPRO                   ;GET R0
                                                EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
                                                        BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
                                                        BNE     1$              ;BR IF YES
                                                        BIT     #SW10,@SWR      ;GOTO NEXT TEST?
                                                        BEQ     2$              ;BR IF NO
                                                        MOV     NEXT,RETURN     ;SET FOR NEXT TEST
                                                1$:     MOV     #STACK,SP       ;RESET SP
                                                        JMP     @RETURN         ;DO SPECIFIED TEST
                                                2$:     RTS     PC              ;RETURN
                                                ERTAB0: .
                                                        .BYTE   6,2
                                                        SAVPC
                                                XTSTN:  .
                                                        .BYTE   3,2
                                                        TSTNO
                                                ;ENTER HERE ON POWER FAILURE
                                                ;---------------------------


                                                .PFAIL:
                                                        MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
                                                        HALT                            ;HALT ON POWER DOWN NORMAL
                                                        BR      .

                                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED


                                                RESTAR:
                                                        MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
```

```
RSX11-S MACY11 27.732A  17-SEP-76  14:08  PAGE 31
[11,1]VAB.P11     GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)


                                                MOV     #STACK,SP       ;RESET THE STACK POINTER
                                                CLR     TEMP            ;READY FOR TIMMER
                                                INC     TEMP            ;PLUS ONE TO THE TIMER!
                                                BNE     .-4             ;BR IF MORE TO GO
                                                TYPE    .MFAIL          ;TYPE THE MESSAGE
                                                CNVRT   .PFTAB          ;TELL WHAT TEST TO RETURN TO.
                                                CLR@    ERRFLG          ;START CLEAN
                                                CLR     LSTERR          ;''''''''''''''''
                                                MSTCLR                  ;START CLEAN UP OF DEVICE
                                                RAMCLR                  ;CLEAR IT ALL!
                                                JMP     DRETURN         ;START DOING THAT TEST AGAIN.

                                PFTAB:          1
                                        002     .BYTE   2,0
                                                ;STNO
                                .DELAY: MOV     R0,-(SP)
                                        MOV     #1,R0
                                        DEC     R0
                                        BNE     .-2
                                        MOV     (SP)+,R0
                                        RTI
                                10:     30.

                                .RAMCLR:
                                        MOV     #MRESET,@DVSCR  ;ISSUE A MASTER CLEAR
                                        MOV     R1,-(SP)        ;SAVE R1 ON THE STACK
                                        MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
                                        MOV     DVSR0,R1        ;GET SECONDARY SEL. REG.
                                        MOV     DVSRA,R4        ;GET SECONDARY REGISTER ACCESS REG.
                                10:     CLR     (R4)            ;READ THE SECONDARY REGISTER.
                                        ADD     #10(BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0)+BIT0,(R1
                                        BNE     1$
                                        MOV     (SP)+,R4        ;RESTORE R4.
                                        MOV     (SP)+,R1        ;RESTORE R1.
                                        RTI

                                .MSTCLR:
                                        MOV     #MRESET,@DVSCR  ;ISSUE MASTER CLEAR.
                                        RTI

                                .ROMCLK:
                                        BIS     #BIT1,@DVSCR
                                        RTI

                                .DATACLK:
                                        MOV     R0,-(SP)
                                        CLR     R0
                                        BIS     #BIT8,@DVLCR
                                10:     MOV     @DVLCR,3$
                                        RORB    (PC)+
                                        BCC     2$
                                        INC     R0
                                        BNE     1$
                                        HLT     0
                                2$:     MOV     (SP)+,R0
                                        RTI
                                3$:     .BLKW 1
```

F03


```
004640  032777  004000  174336  SERV.G: BIT     #4000,@TKCSR    ;RX BUSY?
004646  001374                          BNE     SERV.G          ;BR IF YES
004650  017737  174326  005072          MOV     @SWR,90$        ;SAVE (SWR).
004656  013777  005072  174316  1$:     MOV     90$,@SWR        ;
004664  104402  005052                  TYPE    ,89$            ;
004670  104411  005064                  CNVRT   ,88$            ;
004674  104402  005074                  TYPE    ,91$            ;
004700  105777  174300                  TSTB    @TKCSR          ;WAIT FOR DONE.
004704  100375                          BPL     .-4
004706  017746  174274                  MOV     @TKDBR,-(SP)    ;
004712  042716  000200                  BIC     #BIT7,(SP)      ;
004716  122726  000015                  CMPB    #15,(SP)+       ;
004722  001450                          BEQ     5$              ;
004724  005077  174252                  CLR     @SWR            ;
004730  105777  174254          2$:     TSTB    @TPCSR          ;
004734  100375                          BPL     .-4
004736  016677  177776  174246          MOV     -2(SP),@TPDBR   ;
004744  000241                          CLC                     ;
004746  006177  174230                  ROL     @SWR            ;
004752  006177  174224                  ROL     @SWR            ;
004756  006177  174220                  ROL     @SWR            ;
004762  103736                          BCS     1$              ;ERROR
004764  026627  177776  000060          CMP     -2(SP),#60      ;
004772  002731                          BLT     1$              ;
004774  026627  177776  000067          CMP     -2(SP),#67      ;
005002  003326                          BGT     1$              ;
005004  042766  177770  177776          BIC     #^C<7>,-2(SP)   ;
005012  056677  177776  174162          BIS     -2(SP),@SWR     ;
005020  105777  174160                  TSTB    @TKCSR          ;
005024  100375                          BPL     .-4
005026  017746  174154                  MOV     @TKDBR,-(SP)
005032  042716  000200                  BIC     #BIT7,(SP)
005036  122726  000015                  CMPB    #15,(SP)+
005042  001330                          BNE     2$              ;
005044  104402  005104          5$:     TYPE    ,MCRLF          ;
005050  000207                          RTS     PC

005052  020377  051450  051127  89$:    .ASCIZ  <377>/ (SWR)=/
005060  036451  000057
                                .EVEN
005064  000001                  88$:    1
005066     006     000                  .BYTE   6,0
005070  005072                          90$
005072  000000                  90$:    .WORD   0
005074  036457  000057          91$:    .ASCIZ  /?=/
                                .EVEN
005100  020040  000077          MQM:    .ASCIZ  /  ?/
005104  005015     000          MCRLF:  .ASCIZ  <15><12>
005107     377  053520  020122  MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
005145     377  047105  020104  MEPASS: .ASCIZ  <377>/END PASS DZDVA-B /
005171     377  000122          MR:     .ASCIZ  <377>/R/
005174  050377  047522  051107  MERR2:  .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
005243     377  047111  052523  MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
005267     377  042524  052123  MTSTPC: .ASCIZ  <377>/TEST PC-/
005301     377  047514  046503  MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
```

G03

```
RSX11-S MACY11 27.72(6)  16-SEP-76  14:02  PAGE 33
DRVMS.P11       GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)


                                                MCSRX:  .BLKW   1       ;CSR:
                                                MVECX:  .BLKW   1       ;VEC:
                                                MPASSX: .BLKW   1       ;PASSES:
                                                MERRX:  .BLKW   1       ;ERRORS:
                                                MTSTN:  .BLKW   1       ;TEST NO:
                                                MADTEK: .BLKW   1       ;
                                                MNEW:   .BLKW   1       ;(377) SET SWITCH REG TO DV11'S DESIRED ACTIVE.
                                                MERREG: .BLKW   1       ;(377) MAP OF DV11 STATUS (377
                                                MTEMP:  .BLKW   1       ;(377) MAP OF DV11 STATUS (377
                                                .EVEN
                                                KBUF0:  0
                                                        .BYTE   6.3
                                                        .ASCII  <377>
                                                        .BYTE   6.3
                                                        .ASCII  <377>
                                                .EVEN

                                                ;BUFFERS FOR INPUT-OUTPUT

                                                INBUF:  0
                                                        .BLKW   40
                                                TEMP:   0
                                                        .BLKW   40
                                                MBUF:   0
                                                        .BLKW   40
```

```
                                                        ;
                                                        ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DV11'S
                                                        ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                                        ;AND RUNS THE SPECIFIED DV11'S.  THIS ROUTINE *MUST*
                                                        ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                                        ;SETUP NECESSARY.
                                                        ;
001300  105737  001300          CYCLE:  TSTB    DVACTV          ;ARE ANY DV11'S TO BE TESTED?
001302  001003                          BNE     1$              ;BR IF OK.
001304  104402  005174                  TYPE    .MERR2          ;NO DV11'S SELECTED!!
001306  000000                          HALT                    ;STOP THE SHOW.
001310  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
001312  130037  001304  001300  1$:     BITB    @RUN,DVACTV     ;IS THIS ONE "ACTIVE"
001316  001001                          BNE     2$              ;BR IF GOOD ONE FOUND.
001320  000241                          CLC                     ;CLEAR PROC. CARRY BIT.
001322  106037  001304                  ROLB    @RUN            ;UPDATE POINTER
001326  005037  001304                  ADC     @RUN            ;CATCH CARRY FROM RUN
001332  062737  000004  001306          ADD     #4,CREAM        ;UPDATE ADDRESS POINTER.
001340  022737  001740  001306          CMP     #DV.END,CREAM
001346  001360                          BNE     1$              ;KEEP GOING; NOT ALL TESTED FOR.
001350  012737  001600  001306          MOV     #DV.MAP,CREAM   ;RESET ADDRESS POINTER.
001356  000750                          BR      1$              ;KEEP LOOKING FOR ACTIVE DV11
001360  000241                  2$:     CLC                     ;CLEAR PROC. CARRY.
001362  106137  001304                  ROLB    @RUN            ;UPDATE POINTER.
001366  005137  001304                  ADC     @RUN            ;CATCH CARRY.
001372  013700  001306                  MOV     CREAM,R0        ;GET ADDRESS POINTER.
001376  062700  000004  001306          ADD     #4,CREAM        ;UPDATE.
002004  022737  001740  001306          CMP     #DV.END,CREAM
                                                        ;ALL DONE?
002012  001003                          BNE     3$              ;BR IF NO.
002014  012737  001600  001306          MOV     #DV.MAP,CREAM   ;RESTORE POINTER.
002022  012037  001360          3$:     MOV     (R0)+,@DVSCR    ;LOAD SYSTEM CTRL. REG
002026  012037  001350                  MOV     (R0)+,@DVRVEC   ;LOAD VECTOR
002032  012037  001416                  MOV     (R0)+,LO0.03    ;GET LINE PARAMETERS. 00-03
002036  012037  001426                  MOV     (R0)+,@SYNC2A   ;
002042  012037  001400                  MOV     (R0)+,LO4.07    ;                     04-07
002046  012037  001430                  MOV     (R0)+,@SYNC2B   ;
002052  012037  001420                  MOV     (R0)+,LO8.11    ;                     08-11
002056  012037  001440                  MOV     (R0)+,@SYNC2C   ;
002062  012037  001460                  MOV     (R0)+,L12.15    ;                     12-15
002066  012037  001434                  MOV     (R0)+,@SYNC2D   ;
002072  012700  000002                  MOV     #2,R0           ;SAVE CORE THIS WAY!
002076  013737  001360  001364          MOV     @DVSCR,DVSCRH   ;GET SYS CTRL. REG HIGH BYTE.
002104  005237  001364                  INC     DVSCRH          ;GOT IT.
002110  013737  001364  001366          MOV     DVSCRH,DVRIC    ;GET NXT REC. CHAR REG.
002116  005237  001366                  INC     DVRIC           ;GOT IT
002122  013737  001370  001366          MOV     DVRIC,DVLCR     ;GET LN. PAR.REG.
002130  060037  001370                  ADD     R0,DVLCR        ;GOT IT
002134  013737  001370  001372          MOV     DVLCR,DVSRS     ;GET SEC. REG. SEL. REG.
002142  060037  001372                  ADD     R0,DVSRS        ;GOT IT
002146  013737  001372  001374          MOV     DVSRS,DVSRSH    ;GET HIGH BYTE.
002154  005237  001374                  INC     DVSRSH          ;GOT IT
002160  013737  001374  001376          MOV     DVSRSH,DVSRA    ;SEC. REG. ACCESS.
002166  005237  001376                  INC     DVSRA           ;GOT IT
```

```
DZDVA-B MACY11 27(732)  17-SEP-76  14:02  PAGE 35
DZDVAB.P11      GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

 006166  013737  001376  001400          MOV     DVSRA,DVSFR     ;SPEC. FUN. REG.
 006174  060037  001400                  ADD     R0,DVSFR        ;
 006200  013737  001400  001402          MOV     DVSFR,DVNSR     ;NPR STAT. REG.
 006206  060037  001402                  ADD     R0,DVNSR        ;
 006212  013737  001402  001404          MOV     DVNSR,RESV16    ;RESERVED REG
 006220  060037  001404                  ADD     R0,RESV16       ;

 006224  013737  001352  001354          MOV     DVRVEC,DVRLVL   ;RTY LVL
 006232  060037  001354                  ADD     R0,DVRLVL       ;
 006236  013737  001354  001356          MOV     DVRLVL,DVTVEC   ;TX VEC
 006244  060037  001356                  ADD     R0,DVTVEC       ;
 006250  013737  001356  001360          MOV     DVTVEC,DVTLVL   ;TX LVL
 006256  060037  001360                  ADD     R0,DVTLVL

 006262  012700  001416                  MOV     #L00.03,R0      ;LOAD STAUS 00-03
 006266  012701  001400                  MOV     #MASK.A,R1      ;PREPARE MASK.
 006272  012702  001412                  MOV     #CLK.A,R2       ;PREPARE CLOCKS
 006276  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.

 006302  012700  001420                  MOV     #L04.07,R0      ;LOAD STAUS 00-03
 006306  012701  001407                  MOV     #MASK.B,R1      ;PREPARE MASK.
 006312  012702  001413                  MOV     #CLK.B,R2       ;PREPARE CLOCKS
 006316  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.

 006322  012700  001422                  MOV     #L08.11,R0      ;LOAD STAUS 00-03
 006326  012701  001410                  MOV     #MASK.C,R1      ;PREPARE MASK.
 006332  012702  001414                  MOV     #CLK.C,R2       ;PREPARE CLOCKS
 006336  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.

 006342  012700  001424                  MOV     #L12.15,R0      ;LOAD STAUS 00-03
 006346  012701  001411                  MOV     #MASK.D,R1      ;PREPARE MASK.
 006352  012702  001415                  MOV     #CLK.D,R2       ;PREPARE CLOCKS
 006356  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.
 006362  032777  000002  172612          BIT     #SW01,@ENR
 006370  001445                          BEQ     7$
 006372                          4$:
 006372  005737  000042                  TST     @#42
 006376  001042                          BNE     7$
 006400  104402  005104                  TYPE    .MCRLF
 006404  104403                          INSTR
 006406  005366                          MTSTN
 006410  104405                          PARAM
 006412  000001                          1
 006414  001000                          1000
 006416  001226                          TSTNO
 006420      000                 .BYTE   0
 006421      001                 .BYTE   1
 006422  012700  007256                  MOV     #TST1,R0
 006426  022710                  5$:     CMP     (PC)+,(R0)
 006430  012737                          MOV     (PC)+,@(PC)+
 006432  001015                          BNE     6$
 006434  023760  001226  000002          CMP     TSTNO,2(R0)
 006442  001011                          BNE     6$
 006444  022760  001226  000004          CMP     #TSTNO,4(R0)
 006452  001006                          BNE     6$
 006454  010037  001214                  MOV     R0,RETURN
```

```
 1612   006460   104402   005104                    TYPE    ,MCRLF
 1613   006464   000412                             BR      6$
 1614   006466   005720                    6$:      TST     (R0)+
 1615   006470   020027   031174                    CMP     R0,#TLAST+10
 1616   006474   001365                             BNE     5$
 1617   006476   104402   005100                    TYPE    ,MQM
 1618   006502   000733                             BR      4$
 1619   006504   012737   007256   001314   7$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
 1620   006512   000177   172476            8$:     JMP     @RETURN         ;GO START TESTING.
 1621
 1622   006516   011003                    FIX.00: MOV     (R0),R3         ;GET PARAMETERS.
 1623   006520   042703   176377                    BIC     #^C<1400>,R3    ;CLEAR JUNK.
 1624   006524   005703                             TST     R3              ;TEST FOR EIGHT BITS.
 1625   006526   001004                             BNE     1$              ;BR IF NOT 8 BITS.
 1626   006530   105011                             CLRB    (R1)            ;SET
 1627   006532   112712   000010                    MOVB    #8.,(R2)        ;
 1628   006536   000424                             BR      4$
 1629   006540   022703   000400            1$:     CMP     #400,R3         ;CHECK FOR SEVEN BITS.
 1630   006544   001005                             BNE     2$              ;BR IF NOT 7 BITS.
 1631   006546   112711   000200                    MOVB    #200,(R1)       ;
 1632   006552   112712   000007                    MOVB    #7,(R2)         ;
 1633   006556   000414                             BR      4$
 1634   006560   022703   001000            2$:     CMP     #1000,R3        ;CHECK FOR SIX BITS.
 1635   006564   001005                             BNE     3$              ;BR IF NOT SIX BITS.
 1636   006566   112711   000300                    MOVB    #300,(R1)       ;
 1637   006572   112712   000006                    MOVB    #6.,(R2)        ;
 1638   006576   000404                             BR      4$
 1639   006600   112711   000340            3$:     MOVB    #340,(R1)       ;IF NONE OF THE ABOVE; MUST BE 5 BITS.
 1640   006604   112712   000005                    MOVB    #5.,(R2)        ;
 1641   006610   032710   040000            4$:     BIT     #PARBIT,(R0)    ;PARITY ENABLED?
 1642   006614   001401                             BEQ     5$              ;IF =0; THEN NO PARITY.
 1643   006616   105212                             INCB    (R2)            ;PLUS ONE TO THE CLOCK!
 1644   006620   000207                    5$:     RTS     PC              ;
 1645
 1646                                              ;*ROUTINE USED TO "AUTO SIZE" THE DV11
 1647                                              ;*CSR AND VECTOR.
 1648                                              ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
 1649                                              ;*      ADDRESS RANGE (175000:175400)
 1650                                              ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
 1651                                              ;*      FLOATING VECTOR RANGE (300:770)
 1652                                              ;*
 1653
 1654   006622                    AUTO.SIZE:
 1655   006622   000005                             RESET                   ;INSURE A BUS INIT.
 1656   006624   012702   001500   CSRMAP:  MOV     #DV.MAP,R2      ;LOAD MAP POINTER.
 1657   006630   005022            1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
 1658   006632   022702   001740            CMP     #DV.END,R2      ;ALL DONE?
 1659   006636   001374                     BNE     1$              ;BR IF NO
 1660   006640   105037   001301            CLRB    DVNUM           ;SET OCTAL NUMBER OF DV11'S TO 0
 1661   006644   012702   001500            MOV     #DV.MAP,R2
 1662   006650   012701   175000            MOV     #175000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
 1663   006654   012737   007074   000004   MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
 1664   006662   005711            2$:     TST     (R1)            ;IF DV11 DVSCR S/B 0
 1665   006664   001037                     BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 9 THEN NO DV11
 1666   006666   022761   177777   000012   CMP     #177777,12(R1)  ;IF DV11 THEN DVSFR S/B ALL 1'S ON INIT!
 1667   006674   001033                     BNE     3$              ;BR IF NOT DV11
```

```
1668  006676  005761  000016                  TST     16(R1)          ;IF DV11 THEN RESV16 S/B ALL 0'S
1669  006702  001030                          BNE     3$              ;BR IF NOT DV11
1670                                  ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DV11 CSR ADDRESS.
1671  006704  010122                          MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
1672  006706  005722                          TST     (R2)+           ;POP OVER VECTOR STORE AREA
1673  006710  052722  000226                  BIS     #226,(R2)+      ;SET LINE CARD 1 STAT AND SYNC
1674  006714  052722  000062                  BIS     #62,(R2)+       ;
1675  006720  052722  000226                  BIS     #226,(R2)+      ;SET LINE CARD 2 STAT AND SYNC
1676  006724  052722  000062                  BIS     #62,(R2)+       ;
1677  006730  052722  000226                  BIS     #226,(R2)+      ;SET LINE CARD 3 STAT AND SYNC
1678  006734  052722  000062                  BIS     #62,(R2)+       ;
1679  006740  052722  000226                  BIS     #226,(R2)+      ;SET LINE CARD 4 STAT AND SYNC
1680  006744  052722  000062                  BIS     #62,(R2)+       ;
1681  006750  105237  001301                  INCB    DVNUM           ;UPDATE DEVICE COUNTER
1682  006754  122737  000010  001301          CMPB    #10,DVNUM       ;ARE MAX. NO. OF DEV FOUND?
1683  006762  001405                          BEQ     100$            ;YES DON'T LOOK FOR ANY MORE.
1684  006764  062701  000010          3$:     ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
1685  006770  022701  175400                  CMP     #175400,R1
1686  006774  001332                          BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
1687  006776  012722  177777          100$:   MOV     #177777,(R2)+   ;TERMINATER.
1688  007002  105037  001300                  CLRB    DVACTV
1689  007006  105737  001301                  TSTB    DVNUM           ;WERE ANY DV11'S FOUND AT ALL?
1690  007012  001423                          BEQ     5$              ;ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
1691  007014  113701  001301                  MOVB    DVNUM,R1
1692  007020  110137  001303                  MOVB    R1,SAVNUM       ;SAVE NUMBER OF DEVICES
1693  007024  000241          4$:     CLC
1694  007026  106137  001300                  ROLB    DVACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
1695  007032  105237  001300                  INCB    DVACTV          ;SET THE BIT
1696  007036  005301                          DEC     R1
1697  007040  001371                          BNE     4$              ;BR IF MORE TO GENERATE
1698  007042  012737  000006  000004          MOV     #6,@#4          ;RESTORE TRAP VECTOR
1699  007050  113737  001300  001302          MOVB    DVACTV,SAVACT   ;SAVE ACTIVE REGISTER
1700  007056  000137  007102                  JMP     VECMAP          ;GO FIND THE VECTOR NOW.
1701  007062  104402  005174          5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO DV11'S FOUND.
1702  007066  005000                          CLR     R0              ;MAKE DATA LIGHTS ZERO
1703  007070  000000                          HALT                    ;STOP THE SHOW
1704  007072  000776                          BR      .-2             ;DISABLE CONT. SW.
1705  007074  012716  006764          6$:     MOV     #3$,(SP)        ;ENTERED BY NON-EXISTANT TIME-OUT.
1706  007100  000002                          RTI                     ;RETURN TO MAINSTREAM
1707
1708  007102  012737  000340  000022  VECMAP: MOV     #340,@#22       ;SET IOT TRAP PRIO TO 7
1709  007110  012737  007232  000020          MOV     #4$,@#20        ;SET IOT TRAP VECTOR
1710  007116  012702  001500                  MOV     #DV.MAP,R2      ;SET SOFTWARE POINTER
1711  007122  012700  000300                  MOV     #300,R0         ;FLOATING VECTORS START HERE.
1712  007126  012701  000302                  MOV     #302,R1         ;PC OF IOT INSTR.
1713  007132  010120          1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
1714  007134  012721  000004                  MOV     #4,(R1)+        ;WITH .+2: IOT
1715  007140  022021                          CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
1716  007142  020127  001000                  CMP     R1,#1000
1717  007146  101771                          BLOS    1$              ;BR IF MORE TO FILL
1718  007150  113737  001300  001246          MOVB    DVACTV,TEMP1    ;STORE TEMPORALLY
1719  007156  006037  001246          2$:     ROR     TEMP1           ;BRING OUT A BIT
1720  007162  103034                          BCC     5$              ;BR IF ALL DONE
1721  007164  005037  177776                  CLR     PS              ;ZERO CPU PRIO
1722  007170  012772  001300  000000          MOV     #BIT9+BIT7+BIT6,@(R2)
1723  007176  005000                          CLR     R0              ;ATTEMPT TO FORCE AN INTERUPT
```

```
1724   007200  005200                          INC     R0              ;STALL
1725   007202  001376                          BNE     .-2             ;       FOR TIME TO INTERUPT
1726   007204  052762  000300  000002          BIS     #300,2(R2)      ;NO INTERUPT ASSUME 300 AND FIX DV11 LATER
1727   007212  042772  176777  000000  3$:     BIC     #^C<BIT9>,@(R2)
1728   007220  005072  000000                  CLR     @(R2)
1729   007224  062702  000024                  ADD     #24,R2          ;POP SOFTWARE POINTER
1730   007230  000752                          BR      2$              ;KEEP GOING
1731   007232  051662  000002          4$:     BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
1732   007236  042762  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
1733   007244  022626                          CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
1734   007246  012716  007212                  MOV     #3$,(SP)        ;SET FOR RETURN
1735   007252  000002                          RTI
1736   007254  000207                  5$:     RTS     PC              ;ALL DONE WITH "AUTO SIZING"
1737
```

```
                                        ;********************** 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. NOT
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$:     MOV     (R0),R5         ;LOAD DATA INTO EXPECTED
1809   007532  011304                          MOV     (R3),R4         ;READ THE DV REGISTER
1810   007534  020504                          CMP     R5,R4           ;DOES THE DATA COMPARE
1811   007536  001401                          BEQ     65$             ;BR IF OK
1812   007540  104003                          HLT     3               ;NOW THIS WAS A DUEL ADDRESSING ERROR.
1813   007542  104401                  65$:    SCOP1                   ;SW09=1?
1814   007544  022023                          CMP     (R0)+,(R3)+     ;POP POINTERS
1815   007546  020337  001402                  CMP     R3,DVNSR        ;DON'T DO THE DVNSR
1816   007552  001002                          BNE     5$              ;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.
```

```
                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                        BEQ     1$              ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)         ;CLEAR BIT2
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     2$              ;BR IF OK
                                        HLT     4               ;COMPARISON ERROR
                                2$:     SCOPE                   ;SCOPE THIS TEST

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

                                ;  TEST 4
                                ;--------------
                                TST4:   MOV     #4,TSTNO
                                        MOV     #TST5,NEXT
                                        MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT3,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?
                                        HLT     3               ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)         ;CLEAR BIT3
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     2$              ;BR IF OK
                                        HLT     3               ;COMPARISON ERROR
                                2$:     SCOPE                   ;SCOPE THIS TEST

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

                                ;  TEST 5
                                ;--------------
                                TST5:   MOV     #5,TSTNO
                                        MOV     #TST6,NEXT
                                        MOV     DVSCR,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?
                                        HLT     3               ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)         ;CLEAR BIT4
```

```
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     .+4             ;BR IF OK
                                        HLT                     ;COMPARISON ERROR
                                E8:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 6 ****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BITS, VERIFY BITS WAS SET.
                                        ;*CLEAR BITS, VERIFY BITS WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 6
                                ;--------------
                                ;ST6:    MOV     #6,ISTNO
                                        MOV     #ST10,NEXT
                                        MOV     DVSCR,R0        ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT6,R5        ;SET "EXPECTED".
                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                        BEQ     .+6             ;ARE THEY THE SAME?
                                        HLT                     ;COMPARISON ERROR.
                                ;10:    BIC     R5,(R3)         ;CLEAR BIT6
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     .+4             ;BR IF OK
                                        HLT                     ;COMPARISON ERROR
                                E8:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 7 ****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BITS, VERIFY BITS WAS SET.
                                        ;*CLEAR BITS, VERIFY BITS WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 7
                                ;--------------
                                ;ST7:    MOV     #7,ISTNO
                                        MOV     #ST10,NEXT
                                        MOV     DVSCR,R0        ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT6,R5        ;SET "EXPECTED".
                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                        BEQ     .+6             ;ARE THEY THE SAME?
                                        HLT                     ;COMPARISON ERROR.
                                ;10:    BIC     R5,(R3)         ;CLEAR BIT6
                                        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
                                23:     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
                                ;--------------
                                TST10:  MOV     #10,TSTNO
                                        MOV     #TST11,NEXT
                                        MOV     DVSCR,R3        ;SET REGISTER POINTER INTO R3
                                        MOV     #0,R5           ;SET EXPECTED RESULT
                                        MOV     #BIT7,(R3)      ;ATTEMPT TO SET BIT 7
                                        MOV     (R3),R4         ;READ BACK REGISTER
                                        BEQ     .+4             ;BIT 7 SHOULD NOT BE SET
                                        MTPS    #0              ;DVSCR NOT ALL 0'S
                                        MOV     #BIT7+BIT9,R5   ;SET BIT 7+BIT9 IN THE DVSCR
                                        MOV     R5,(R3)         ;
                                        MOV     (R3),R4         ;READ REGISTER
                                        CMP     R5,R4           ;IS BIT 9 AND BIT 7 SET?
                                        BEQ     .+6             ;BR IF OK
                                        MTPS    #0              ;DVSCR ERROR
                                        BIC     #BIT7,R5        ;CLEAR BIT 7
                                        BIC     #BIT7,(R3)      ;DITTO
                                        MOV     (R3),R4         ;READ REGISTER
                                        CMP     R5,R4           ;COMPARE OK?
                                        BEQ     .+6             ;BR IF YES
                                        MTPS    #0              ;DVSCR ERROR
                                        SCOPE                   ;SCOPE THIS TEST


                                ;************************ TEST 11 ******************************
                                ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                ;*SET BIT8, VERIFY BIT8 WAS SET.
                                ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
                                ;;*************************************************************

                                ;  TEST 11
                                ;--------------
                                TST11:  MOV     #11,TSTNO
                                        MOV     #TST12,NEXT
                                        MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT8,R5        ;SET "EXPECTED".
                                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                                        BEQ     .+6             ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                        BIC     R5,(R3)         ;CLEAR BIT8
                                        MOV     (R3),R4         ;READ THE REGISTER.
```

```
                                        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 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     DVSCR,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     1$              ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)         ;CLEAR BIT9
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     2$              ;BR IF OK
                                        HLT     3               ;COMPARISON ERROR
                                2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 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     DVSCR,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     1$              ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)         ;CLEAR BIT10
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     2$              ;BR IF OK
                                        HLT     3               ;COMPARISON ERROR
```

F04


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

                                ;  TEST 14
                                ;--------------
                                TST14:  MOV     #14,TSTNO
                                        MOV     #TST15,NEXT
                                        MOV     DVSCR,R3                ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT12,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?
                                        HLT     3                       ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)                 ;CLEAR BIT12
                                        MOV     (R3),R4                 ;READ THE REGISTER.
                                        CLR     R5                      ;SET "EXPECTED"
                                        CMP     R5,R4                   ;R5=GOOD; R4=?
                                        BEQ     2$                      ;BR IF OK
                                        HLT     3                       ;COMPARISON ERROR
                                2$:     SCOPE                           ;SCOPE THIS TEST


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

                                ;  TEST 15
                                ;--------------
                                TST15:  MOV     #15,TSTNO
                                        MOV     #TST16,NEXT
                                        MOV     DVSCR,R3                ;SET REGISTER TO BE TESTED.
                                        MOV     #BIT13,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?
                                        HLT     3                       ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)                 ;CLEAR BIT13
                                        MOV     (R3),R4                 ;READ THE REGISTER.
                                        CLR     R5                      ;SET "EXPECTED"
                                        CMP     R5,R4                   ;R5=GOOD; R4=?
                                        BEQ     2$                      ;BR IF OK
                                        HLT     3                       ;COMPARISON ERROR
                                2$:     SCOPE                           ;SCOPE THIS TEST


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

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

                                ;  TEST 16
                                ;--------------
 0130  016604  016737  000016  001606  TST16:  MOV     #16,TSTNO
 0131  016606  016737  016660  001616          MOV     #TST17,NEXT
 0132  016614  016703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
 0133  016620  012713  000011                  MOV     #BIT3+BIT0,R5   ;SET "EXPECTED"
 0134  016624  010523                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 0135  016626  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 0136  016630  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 0137  016632  001402                          BEQ     1$              ;ARE THEY THE SAME?
 0138  016634  000403                          BR      2$              ;COMPARISON ERROR.
 0139  016636  105013                  1$:     CLRB    (R3)            ;CLEAR BIT3+BIT0
 0140  016640  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 0141  016642  005005                          CLR     R5              ;SET "EXPECTED"
 0142  016644  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
 0143  016646  001402                          BEQ     3$              ;BR IF OK
 0144  016650  000403                          BR      4$              ;COMPARISON ERROR
 0145  016652  104400                  2$:     SCOPE                   ;SCOPE THIS TEST

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

                                ;  TEST 17
                                ;--------------
 0146  016654  016737  000017  001606  TST17:  MOV     #17,TSTNO
 0147  016662  016737  016704  001616          MOV     #TST20,NEXT
 0148  016670  016703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER INTO R3
 0149  016674  005006                          CLR     R6              ;SET EXPECTED RESULT
 0150  016676  012713  100000                  MOV     #BIT15,(R3)     ;ATTEMPT TO SET BIT 15
 0151  016702  011304                          MOV     (R3),R4         ;READ BACK REGISTER
 0152  016704  001402                          BEQ     1$              ;BIT 15 SHOULD NOT BE SET
 0153  016706  000403                          BR      2$              ;DVSCR NOT ALL 0'S
                                        ;
 0154  016710  012705  101000          1$:     MOV     #BIT15+BIT9,R5  ;SET BIT 15+BIT9 IN THE DVSCR
 0155  016714  105013                          CLRB    (R3)            ;
 0156  016716  011304                          MOV     (R3),R4         ;READ REGISTER
 0157  016720  020504                          CMP     R5,R4           ;IS BIT 9 AND BIT 15 SET?
 0158  016722  001402                          BEQ     3$              ;BR IF OK
 0159  016724  000403                          BR      4$              ;DVSCR ERROR
 0160  016726  012713  100000          2$:     MOV     #BIT15,R5       ;CLEAR BIT 15
 0161  016732  012703  100000                  MOV     #BIT15,(R3)     ;DITTO
 0162  016736  011304                          MOV     (R3),R4         ;READ REGISTER
 0163  016740  020504                          CMP     R5,R4           ;COMPARE OK?
 0164  016742  001402                          BEQ     3$              ;BR IF YES
 0165  016744  000403                          BR      4$              ;DVSCR ERROR
 0166  016746  104400                  3$:     SCOPE                   ;SCOPE THIS TEST
```

```
                                        ;************************ TEST 20 ************************************
                                        ;*TEST THAT BIT8 IN DVSCR CLEARS.
                                        ;*BIT7 OF DVSCR.
                                        ;;*********************************************************************

                                ;  TEST 20
                                ;--------------
011054  012737  000020  001226  TST20:  MOV     #20,TSTNO
011062  012737  011130  001216          MOV     #TST21,NEXT
011070  013703  001346                  MOV     DVSCR,R3        ;GET POINTER
011074  012704  000400                  MOV     #BIT8,R4        ;SET GOOD
011100  010413                          MOV     R4,(R3)         ;SET BIT7
011102  010423                          MOV     R4,(R3)+        ;SET BIT8
011104  005723  177776  177776          TST     @DVRIC,@DVRIC   ;WASTE TIME
011110  011304                          MOV     (R3),R4         ;READ SCR
011112  020504                          CMP     R5,R4           ;BIT8 ONLY SET?
011114  001403                          BEQ     1$              ;BR IF YES
011116  104003                          HALT    3               ;BIT8 NOT ONLY BIT SET OR 8 ISN'T SET!
011120  005013                  1$:     CLR     (R3)            ;LEAVE REG=0
011122  104400                          SCOPE                   ;SCOPE

                                        ;************************ TEST 21 ************************************
                                        ;*RECEIVER INTERUPT CHARACTER REGISTER TEST
                                        ;*TEST 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
                                ;--------------
011124  012737  000021  001226  TST21:  MOV     #21,TSTNO
011132  012737  011170  001216          MOV     #TST22,NEXT
011140  013703  001366                  MOV     DVRIC,R3
                                                                ;GET THE REGISTER.
011144  005005                          CLR     R5              ;SET EXPECTED (ZERO)
011146  012713  177777                  MOV     #-1,(R3)        ;WRITE REGISTER WITH ALL 1'S
011152  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011154  020504                          CMP     R5,R4           ;IS THE REGISTER EQUAL TO ZERO.
011156  001403                          BEQ     1$              ;BR IF OK.
011160  104003                          HALT    3               ;REGISTER NOT ZERO.
011162  104400                  1$:     SCOPE                   ;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
                                ;--------------
011164  012737  000022  001226  TST22:  MOV     #22,TSTNO
011172  012737  011216  001216          MOV     #TST23,NEXT
011200  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
011112  012705  001000                MOV     #BIT9,R5        ;SET "EXPECTED ".
011116  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
011120  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011122  042704  000063                BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011126  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011130  001401                        BEQ     1$              ;ARE THEY THE SAME?
011132  104003                        HLT     3               ;COMPARISON ERROR.
011134  040513                1$:     BIC     R5,(R3)         ;CLEAR BIT9
011136  011304                        MOV     (R3),R4         ;READ THE REGISTER.
011140  042704  000063                BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011144  005005                        CLR     R5              ;SET "EXPECTED"
011146  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
011150  001401                        BEQ     2$              ;BR IF OK
011152  104003                        HLT     3               ;COMPARISON ERROR
011154  104400                2$:     SCOPE                   ;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
                                ;--------------
011156  012737  000023  001226  TST23:  MOV     #23,TSTNO
011164  012737  011240  001216         MOV     #TST24,NEXT
011172  013703  001370                 MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
011176  012705  002000                 MOV     #BIT10,R5       ;SET "EXPECTED ".
011202  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
011204  011304                         MOV     (R3),R4         ;READ THE REGISTER.
011206  042704  000063                 BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011212  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011214  001401                         BEQ     1$              ;ARE THEY THE SAME?
011216  104003                         HLT     3               ;COMPARISON ERROR.
011220  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT10
011222  011304                         MOV     (R3),R4         ;READ THE REGISTER.
011224  042704  000063                 BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011230  005005                         CLR     R5              ;SET "EXPECTED"
011232  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
011234  001401                         BEQ     2$              ;BR IF OK
011236  104003                         HLT     3               ;COMPARISON ERROR
011240  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
                                ;--------------
011242  012737  000024  001226  TST24:  MOV     #24,TSTNO
011250  012737  011326  001216         MOV     #TST25,NEXT
011256  013703  001370                 MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
011262  012705  004000                  MOV     #BIT11,R5       ;SET "EXPECTED ".
011266  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011270  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011272  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011276  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011300  001401                          BEQ     1$              ;ARE THEY THE SAME?
011302  104003                          HLT     3               ;COMPARISON ERROR.
011304  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT11
011306  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011310  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011314  005005                          CLR     R5              ;SET "EXPECTED"
011316  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011320  001401                          BEQ     2$              ;BR IF OK
011322  104003                          HLT     3               ;COMPARISON ERROR
011324  104400                  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  000063                  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.
```

```
 2354  011432  012705  020000                    MOV     #BIT13,R5       ;SET "EXPECTED ".
 2355  011436  010513                            MOV     R5,(R3)         ;WRITE THE REGISTER.
 2356  011440  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2357  011442  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2358  011446  020504                            CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2359  011450  001401                            BEQ     1$              ;ARE THEY THE SAME?
 2360  011452  104003                            HLT     3               ;COMPARISON ERROR.
 2361  011454  040513                    1$:     BIC     R5,(R3)         ;CLEAR BIT13
 2362  011456  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2363  011460  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2364  011464  005005                            CLR     R5              ;SET "EXPECTED"
 2365  011466  020504                            CMP     R5,R4           ;R5=GOOD; R4=?
 2366  011470  001401                            BEQ     2$              ;BR IF OK
 2367  011472  104003                            HLT     3               ;COMPARISON ERROR
 2368  011474  104400                    2$:     SCOPE                   ;SCOPE THIS TEST
 2369
 2370
 2371                                            ;********************** TEST 27 ****************************
 2372                                            ;*LINE CONTROL REGISTER READ/WRITE TEST.
 2373                                            ;*SET BIT14, VERIFY BIT14 WAS SET.
 2374                                            ;*CLEAR BIT14, VERIFY BIT14 WAS CLEARED.
 2375                                            ;:************************************************************
 2376
 2377                                    ;  TEST 27
 2378                                    ;--------------
 2379  011476  012737  000027  001226    TST27:  MOV     #27,TSTNO
 2380  011504  012737  011562  001216            MOV     #TST30,NEXT
 2381  011512  013703  001370                    MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
 2382  011516  012705  040000                    MOV     #BIT14,R5       ;SET "EXPECTED ".
 2383  011522  010513                            MOV     R5,(R3)         ;WRITE THE REGISTER.
 2384  011524  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2385  011526  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2386  011532  020504                            CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2387  011534  001401                            BEQ     1$              ;ARE THEY THE SAME?
 2388  011536  104003                            HLT     3               ;COMPARISON ERROR.
 2389  011540  040513                    1$:     BIC     R5,(R3)         ;CLEAR BIT14
 2390  011542  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2391  011544  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2392  011550  005005                            CLR     R5              ;SET "EXPECTED"
 2393  011552  020504                            CMP     R5,R4           ;R5=GOOD; R4=?
 2394  011554  001401                            BEQ     2$              ;BR IF OK
 2395  011556  104003                            HLT     3               ;COMPARISON ERROR
 2396  011560  104400                    2$:     SCOPE                   ;SCOPE THIS TEST
 2397
 2398
 2399                                            ;********************** TEST 30 ****************************
 2400                                            ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
 2401                                            ;*SET BIT0, VERIFY BIT0 WAS SET.
 2402                                            ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
 2403                                            ;:************************************************************
 2404
 2405                                    ;  TEST 30
 2406                                    ;--------------
 2407  011562  012737  000030  001226    TST30:  MOV     #30,TSTNO
 2408  011570  012737  011636  001216            MOV     #TST31,NEXT
 2409  011576  013703  001372                    MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
```

```
011602  012705  000001                  MOV     #BIT0,R5        ;SET "EXPECTED ".
011606  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011610  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011612  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011614  001401                          BEQ     1$              ;ARE THEY THE SAME?
011616  104003                          HLT     3               ;COMPARISON ERROR.
011620  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT0
011622  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011624  005005                          CLR     R5              ;SET "EXPECTED"
011626  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011630  001401                          BEQ     2$              ;BR IF OK
011632  104003                          HLT     3               ;COMPARISON ERROR
011634  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.
```

```
                                        BEQ     1$              ;ARE THEY THE SAME?
                                        HLT     3               ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)         ;CLEAR BIT2
                                        MOV     (R3),R4         ;READ THE REGISTER.
                                        CLR     R5              ;SET "EXPECTED"
                                        CMP     R5,R4           ;R5=GOOD; R4=?
                                        BEQ     2$              ;BR IF OK
                                        HLT     3               ;COMPARISON ERROR
                                2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 33 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT3, VERIFY BIT3 WAS SET.
                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
                                        ;:***********************************************************


                                ;  TEST 33
                                ;--------------
011766  012737  000033  001226  TST33:  MOV     #33,TSTNO
011774  012737  012042  001216          MOV     #TST34,NEXT
012002  013703  001372                  MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
012006  012705  000010                  MOV     #BIT3,R5        ;SET "EXPECTED "
012012  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
012014  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012016  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
012020  001401                          BEQ     1$              ;ARE THEY THE SAME?
012022  104003                          HLT     3               ;COMPARISON ERROR.
012024  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT3
012026  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012030  005005                          CLR     R5              ;SET "EXPECTED"
012032  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
012034  001401                          BEQ     2$              ;BR IF OK
012036  104003                          HLT     3               ;COMPARISON ERROR
012040  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 34 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT8, VERIFY BIT8 WAS SET.
                                        ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
                                        ;:***********************************************************


                                ;  TEST 34
                                ;--------------
012042  012737  000034  001226  TST34:  MOV     #34,TSTNO
012050  012737  012116  001216          MOV     #TST35,NEXT
012056  013703  001372                  MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
012062  012705  000400                  MOV     #BIT8,R5        ;SET "EXPECTED "
012066  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
012070  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012072  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
012074  001401                          BEQ     1$              ;ARE THEY THE SAME?
012076  104003                          HLT     3               ;COMPARISON ERROR.
012100  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT8
012102  011304                          MOV     (R3),R4         ;READ THE REGISTER.
```

```
                                        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 35 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT9, VERIFY BIT9 WAS SET.
                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
                                        ;:************************************************************

                                ;  TEST 35
                                ;--------------
012116  012737  000035  001226  TST35:  MOV     #35,TSTNO
012124  012737  012172  001216          MOV     #TST36,NEXT
012132  013703  001372                  MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
012136  012705  001000                  MOV     #BIT9,R5        ;SET "EXPECTED ".
012142  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
012144  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012146  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
012150  001401                          BEQ     1$              ;ARE THEY THE SAME?
012152  104003                          HLT     3               ;COMPARISON ERROR.
012154  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT9
012156  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012160  005005                          CLR     R5              ;SET "EXPECTED"
012162  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
012164  001401                          BEQ     2$              ;BR IF OK
012166  104003                          HLT     3               ;COMPARISON ERROR
012170  104400                  2$:     SCOPE                   ;SCOPE THIS TEST

                                        ;************************ TEST 36 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT10, VERIFY BIT10 WAS SET.
                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
                                        ;:************************************************************

                                ;  TEST 36
                                ;--------------
012172  012737  000036  001226  TST36:  MOV     #36,TSTNO
012200  012737  012246  001216          MOV     #TST37,NEXT
012206  013703  001372                  MOV     DVSR5,R3        ;SET REGISTER TO BE TESTED.
012212  012705  002000                  MOV     #BIT10,R5       ;SET "EXPECTED ".
012216  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
012220  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012222  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
012224  001401                          BEQ     1$              ;ARE THEY THE SAME?
012226  104003                          HLT     3               ;COMPARISON ERROR.
012230  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT10
012232  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012234  005005                          CLR     R5              ;SET "EXPECTED"
012236  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
012240  001401                          BEQ     2$              ;BR IF OK
012242  104003                          HLT     3               ;COMPARISON ERROR
```

```
                                                ;SCOPE THIS TEST

                                        ;********************** TEST 37 ***********************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT11, VERIFY BIT11 WAS SET.
                                        ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
                                        ;;*****************************************************************

                                ;  TEST 37
                                ;--------------
                                TST37:  MOV     #37,TSTNO
                                        MOV     #TST40,NEXT
                                        MOV     CVSR0,R3                ;SET REGISTER TO BE TESTED.
                                        MOV     #177777,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.
                                10$:    MOV     R0,(R3)                 ;CLEAR BIT11
                                        MOV     (R3),R4                 ;READ THE REGISTER.
                                        CMP     R0,R4                   ;SET "EXPECTED"
                                        BEQ     1$                      ;R0=GOOD; R4=?
                                        HALT                            ;BR IF OK
                                        BR      10$                     ;COMPARISON ERROR
                                1$:     SCOPE                           ;SCOPE THIS TEST

                                        ;********************** TEST 40 ***********************************
                                        ;*SECONDARY REGISTER ACCESS REG. READ/WRITE TEST.
                                        ;*SET EACH INDIVIDUAL BIT; VERIFY EACH INDIVIDUAL BIT SET.
                                        ;*CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.
                                        ;;*****************************************************************

                                ;  TEST 40
                                ;--------------
                                TST40:  MOV     #40,TSTNO
                                        MOV     #TST41,NEXT
                                        MOV     CVSR0,R3                ;SET CSR POINTER INTO R3
                                        MOV     #177777,R5              ;SET EXPECTED TO ALL 1'S
                                        MOV     R5,(R3)                 ;WRITE REGISTER WITH EXPECTED DATA
                                        MOV     (R3),R4                 ;READ THE REGISTER
                                        CMP     R5,R4                   ;WAS READ EQUAL TO WRITTEN?
                                        BEQ     1$                      ;BR IF YES
                                        HALT                            ;PRIMARY REGISTER READ WRITE TEST
                                10$:    CLR     R5                      ;SET DATA TO ZERO
                                        MOV     R5,(R3)                 ;WRITE REGISTER
                                        MOV     (R3),R4                 ;READ REGISTER
                                        CMP     R5,R4                   ;REGISTER WRITTEN-OK?
                                        BEQ     1$                      ;BR IF YES
                                        HALT                            ;PRIMARY REGISTER DATA ERROR
                                100$:   MOV     #1,R5                   ;SET FOR "FLOATING 1"
                                        MOV     R5,(R3)                 ;WRITE DATA
                                        MOV     (R3),R4                 ;READ DATA
```

```
                                        CMP     R5,R4           ;DATA OK?
                                        BEQ     4$              ;BR IF YES
                                        HLT     3               ;DATA ERROR
                                4$:     CLC                     ;ZERO CPU CARRY
                                        ROL     R5              ;UPDATE DATA
                                        BNE     3$              ;BR IF MORE TO GO
                                        CLR     (R3)            ;LEAVE REG ALL 0'S
                                        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
                                ;--------------
                                TST41:  MOV     #41,ISTNO
                                        MOV     #TST42,NEXT
                                        MOV     @#0174,R3       ;SET SOURCE SEL
                                        MOV     DVSFR,R3                ;SET CSR POINTER INTO R3
                                        MOV     #1,R5           ;SET EXPECTED TO ALL 1'S
                                        MOV     R5,(R3)         ;WRITE REGISTER WITH EXPECTED DATA
                                        MOV     (R3),R4         ;READ THE REGISTER
                                        CMP     R5,R4           ;WAS READ EQUAL TO WRITTEN??
                                        BEQ     1$              ;BR IF YES
                                        HLT     3               ;PRIMARY REGISTER READ/WRITE TEST
                                1$:     CLR     R5              ;SET DATA TO ZERO
                                        MOV     R5,(R3)         ;WRITE REGISTER
                                        MOV     (R3),R4         ;READ REGISTER
                                        CMP     R5,R4           ;REGISTER WRITTEN OK?
                                        BEQ     2$              ;BR IF YES
                                        HLT     3               ;PRIMARY REGISTER DATA ERROR
                        000001          MOV     #1,R5           ;SET FOR "FLOATING 1"
                                3$:     MOV     R5,(R3)         ;WRITE DATA
                                        MOV     (R3),R4         ;READ DATA
                                        CMP     R5,R4           ;DATA OK?
                                        BEQ     4$              ;BR IF YES
                                        HLT     3               ;DATA ERROR
                                4$:     CLC                     ;ZERO CPU CARRY
                                        ROL     R5              ;UPDATE DATA
                                        BNE     3$              ;BR IF MORE TO GO
                                        CLR     (R3)            ;LEAVE REG ALL 0'S
                                        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.
                                        ;;*************************************************************
```

```
                                        ;  TEST 42
                                        ;--------------
018504  012737  000042  001836  TST42:  MOV     #42,TSTNO
018512  012737  018550  001816          MOV     #TST43,NEXT
018520  013703  001402                  MOV     DVNBR,R3        ;GET THE REGISTER.
018524  011305                          MOV     (R3),R5         ;READ THE REGISTER INTO R5
018526  010504                          MOV     R5,R4           ;SAVE REG. INTO R4
018530  005104                          COM     R4              ;MAKE R4 OPPOSITE TO REGISTER
018532  010413                          MOV     R4,(R3)         ;WRITE THE REGISTER WITH THE COMPLIMENT
018534  011304                          MOV     (R3),R4         ;READ THE REGISTER.
018536  020405                          CMP     R5,R4           ;IS THE REGISTER EQUAL TO ZERO.
018540  001402                          BEQ     1$              ;OK IF OK.
018542  104000                          HALT                    ;REGISTER NOT ZERO.
018544  104400                  1$:     SCOPE                   ;SCOPE THIS TEST.

                                        ;********************** TEST 43 ****************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT0, VERIFY BIT0 WAS SET.
                                        ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
                                        ;;************************************************************

                                        ;  TEST 43
                                        ;--------------
018546  012737  000043  001836  TST43:  MOV     #43,TSTNO
018554  012737  018610  001816          MOV     #TST44,NEXT
018562  013703  001402                  MOV     DVNBR,R3        ;GET REGISTER TO BE TESTED.
018566  012705  000001                  MOV     #BIT0,R5        ;SET "EXPECTED ".
018572  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
018574  011304                          MOV     (R3),R4         ;READ THE REGISTER.
018576  020405                          CMP     R4,R5           ;R4=GOOD; R4=UNKNOWN.
018600  001402                          BEQ     1$              ;ARE THEY THE SAME?
018602  104000                          HALT                    ;COMPARISON ERROR.
018604  005003                          CLR     R3              ;CLEAR BIT0
018606  010314                  1$:     MOV     R3,(R4)         ;READ THE REGISTER.
018610  011304                          MOV     (R3),R4         ;SET "EXPECTED"
018612  020405                          CMP     R4,R5           ;R4=GOOD; R4=?
018614  001402                          BEQ     2$              ;IF OK
018616  104000                          HALT                    ;COMPARISON ERROR
018620  104400                  2$:     SCOPE                   ;SCOPE THIS TEST

                                        ;********************** TEST 44 ****************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT1, VERIFY BIT1 WAS SET.
                                        ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
                                        ;;************************************************************

                                        ;  TEST 44
                                        ;--------------
018622  012737  000044  001836  TST44:  MOV     #44,TSTNO
018630  012737  018672  001816          MOV     #TST45,NEXT
018636  013703  001402                  MOV     REGDV16,R3      ;SET REGISTER TO BE TESTED.
018642  012705  000002                  MOV     #BIT1,R5        ;SET "EXPECTED ".
```

```
RSX11-S MACY11 27(732)  17-SEP-76  14:02  PAGE 57
DZVAB.P11       DV11 DEVICE DIAGNOSTICS.    COPYRIGHT 1975  DIGITAL EQUIP. CORP.

                                                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$:     BIC     R5,(R3)         ;CLEAR BIT1
                                                MOV     (R3),R4         ;READ THE REGISTER.
                                                CLR     R5              ;SET "EXPECTED"
                                                CMP     R5,R4           ;R5=GOOD; R4=?
                                                BEQ     2$              ;BR IF OK
                                                HALT                    ;COMPARISON ERROR
                                        2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 45 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT2, VERIFY BIT2 WAS SET.
                                        ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
                                        ;;***********************************************************

                                        ;  TEST 45
                                        ;--------------
                                        TST45:  MOV     #45,TSTNO
                                                MOV     #TST47,NEXT
                                                MOV     #DSV16,R3       ;SET REGISTER TO BE TESTED.
                                                MOV     #BIT2,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$:     BIC     R5,(R3)         ;CLEAR BIT2
                                                MOV     (R3),R4         ;READ THE REGISTER.
                                                CLR     R5              ;SET "EXPECTED"
                                                CMP     R5,R4           ;R5=GOOD; R4=?
                                                BEQ     2$              ;BR IF OK
                                                HALT                    ;COMPARISON ERROR
                                        2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 46 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT3, VERIFY BIT3 WAS SET.
                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
                                        ;;***********************************************************

                                        ;  TEST 46
                                        ;--------------
                                        TST46:  MOV     #46,TSTNO
                                                MOV     #TST47,NEXT
                                                MOV     #DSV16,R3       ;SET REGISTER TO BE TESTED.
                                                MOV     #BIT3,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?
```

```
                                        HLT     3                       ;COMPARISON ERROR.
                                1$:     BIC     R5,(R3)                 ;CLEAR BIT3
                                        MOV     (R3),R4                 ;READ THE REGISTER.
                                        CLR     R5                      ;SET "EXPECTED"
                                        CMP     R5,R4                   ;R5=GOOD; R4=?
                                        BEQ     2$                      ;BR IF OK
                                        HLT     3                       ;COMPARISON ERROR
                                2$:     SCOPE                           ;SCOPE THIS TEST


                                        ;************************ TEST 47 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT4, VERIFY BIT4 WAS SET.
                                        ;*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
                                        ;;***********************************************************

                                ;  TEST 47
                                ;--------------
013056  012737  000047  001226  TST47:  MOV     #47,TSTNO
013064  012737  013130  001216          MOV     #TST50,NEXT
013072  013703  001434                  MOV     RESV16,R3               ;SET REGISTER TO BE TESTED.
013076  012705  000020                  MOV     #BIT4,R5                ;SET "EXPECTED"
013102  010513                          MOV     R5,(R3)                 ;WRITE THE REGISTER.
013104  011304                          MOV     (R3),R4                 ;READ THE REGISTER.
013106  020504                          CMP     R5,R4                   ;R5=GOOD; R4=UNKNOWN.
013110  001401                          BEQ     1$                      ;ARE THEY THE SAME?
013112  104003                          HLT     3                       ;COMPARISON ERROR.
013114  040513                  1$:     BIC     R5,(R3)                 ;CLEAR BIT4
013116  011304                          MOV     (R3),R4                 ;READ THE REGISTER.
013120  005005                          CLR     R5                      ;SET "EXPECTED"
013122  020504                          CMP     R5,R4                   ;R5=GOOD; R4=?
013124  001401                          BEQ     2$                      ;BR IF OK
013126  104003                          HLT     3                       ;COMPARISON ERROR
013130  104400                  2$:     SCOPE                           ;SCOPE THIS TEST


                                        ;************************ TEST 50 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BITS, VERIFY BITS WAS SET.
                                        ;*CLEAR BITS, VERIFY BITS WAS CLEARED.
                                        ;;***********************************************************

                                ;  TEST 50
                                ;--------------
013132  012737  000050  001226  TST50:  MOV     #50,TSTNO
013140  012737  013206  001216          MOV     #TST51,NEXT
013146  013703  001434                  MOV     RESV16,R3               ;SET REGISTER TO BE TESTED.
013152  012705  000340                  MOV     #BITS,R5                ;SET "EXPECTED"
013156  010513                          MOV     R5,(R3)                 ;WRITE THE REGISTER.
013160  011304                          MOV     (R3),R4                 ;READ THE REGISTER.
013162  020504                          CMP     R5,R4                   ;R5=GOOD; R4=UNKNOWN.
013164  001401                          BEQ     1$                      ;ARE THEY THE SAME?
013166  104003                          HLT     3                       ;COMPARISON ERROR.
013170  040513                  1$:     BIC     R5,(R3)                 ;CLEAR BITS
013172  011304                          MOV     (R3),R4                 ;READ THE REGISTER.
013174  005005                          CLR     R5                      ;SET "EXPECTED"
```

```
013176  005504                  CMP     R5,R4           ;R5=GOOD; R4=?
013200  001401                  BEQ     2$              ;BR IF OK
013202  104003                  HLT     3               ;COMPARISON ERROR
013204  104430          2$:     SCOPE                   ;SCOPE THIS TEST


                        ;********************** TEST 51 ****************************
                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                        ;*SET BIT6, VERIFY BIT6 WAS SET.
                        ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
                        ;;*************************************************************

                        ;  TEST 51
                        ;---------------
013206  012737  000051  001226  TST51:  MOV     #51,TSTNO
013214  012737  013260  001216          MOV     #TST52,NEXT
013222  012703  014000                  MOV     #REGV16,R3      ;SET REGISTER TO BE TESTED.
013226  012705  000100                  MOV     #BIT6,R5        ;SET "EXPECTED".
013232  010513                  1$:     MOV     R5,(R3)         ;WRITE THE REGISTER.
013234  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013236  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
013240  001401                          BEQ     2$              ;ARE THEY THE SAME?
013242  104003                          HLT     3               ;COMPARISON ERROR.
013244  042513                  2$:     BIC     R5,(R3)         ;CLEAR BIT6
013246  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013250  020504                          CMP     R5,R4           ;SET "EXPECTED"
013252  001401                          BEQ     3$              ;R5=GOOD; R4=?
013254  104003                          HLT     3               ;BR IF OK
013256  104430          3$:     SCOPE                   ;COMPARISON ERROR
                                                        ;SCOPE THIS TEST


                        ;********************** TEST 52 ****************************
                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                        ;*SET BIT7, VERIFY BIT7 WAS SET.
                        ;*CLEAR BIT7, VERIFY BIT7 WAS CLEARED.
                        ;;*************************************************************

                        ;  TEST 52
                        ;---------------
013260  012737  000052  001216  TST52:  MOV     #52,TSTNO
013266  012737  013320  001216          MOV     #TST53,NEXT
013274  012703  014000                  MOV     #REGV16,R3      ;SET REGISTER TO BE TESTED.
013300  012705  000200                  MOV     #BIT7,R5        ;SET "EXPECTED".
013304  010513                  1$:     MOV     R5,(R3)         ;WRITE THE REGISTER.
013306  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013310  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
013312  001401                          BEQ     2$              ;ARE THEY THE SAME?
013314  104003                          HLT     3               ;COMPARISON ERROR.
013316  042513                  2$:     BIC     R5,(R3)         ;CLEAR BIT7
013320  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013322  020504                          CMP     R5,R4           ;SET "EXPECTED"
013324  001401                          BEQ     3$              ;R5=GOOD; R4=?
013326  104003                          HLT     3               ;BR IF OK
013330  104430          3$:     SCOPE                   ;COMPARISON ERROR
                                                        ;SCOPE THIS TEST
```

```
                                        ;+++++++++++++++++++++++++ TEST 53 ++++++++++++++++++++++++++++++
                                        ;*TEST OF THE BYTE OPERATIONS FOR THE DV11
                                        ;*SYSTEM CONTROL REG AND THE SECONDARY REG SEL.
                                        ;*THE TEST WILL CLEAR DVSCR AND THE WRITE (LOW BYTE)
                                        ;*BIT3. THEN VERIFY ONLY BIT3 IS SET. THEN THE
                                        ;*TEST WILL WRITE BIT 8(HIGH BYTE) AND VERFIY THAT
                                        ;*BIT8 AND BIT3 ARE SET. THE EXACT PROCEEDURE
                                        ;*WILL BE USED ON THE DVSRS REGISTER.
                                        ;;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                ;  TEST 53
                                ;-------------
013036  012737  000053  001006  TST53:  MOV     #53,ISTNO
013044  013767  013470  001016          MOV     #TST54,NEXT
013052  013703  001130                  MOV     DVSCR,R3        ;SET DVSCR POINTER
013056  005013                  1$:     CLR     (R3)            ;MAKE SURE IT =0
013060  012705  000010                  MOV     #BIT3,R5        ;LOAD EXPECTED RESULTS
013064  110513                  2$:     MOVB    R5,(R3)         ;"WRITE" BYTE (LOW) BIT3
013066  011304                  3$:     MOV     (R3),R4         ;READ (WORD) RESULT
013070  020504                  4$:     CMP     R5,R4           ;MAKE SURE ONLY BIT3 IS SET
013072  001403                  5$:     BEQ     6$              ;BR IF OK
013074  104000                  7$:     HLT     0               ;DVSCR WRONG
013076  012705  000410          6$:     MOV     #BIT8+BIT3,R5   ;SET EXPECTED DATA
013102  110513  000001          8$:     MOVB    #BIT0,1(R3)     ;"WRITE" BYTE (HIGH) BIT0 (BIT8 OF WORD)
013106  011304                  9$:     MOV     (R3),R4         ;READ (WORD) RESULT
013110  020504                  10$:    CMP     R5,R4           ;OK?
013112  001403                  11$:    BEQ     12$             ;
013114  104000                  13$:    HLT     0               ;DVSCR WRONG
013116  005013                  12$:    CLR     (R3)            ;LEAVE REGISTERED CLEARED
013120  013703  001136                  MOV     DVSRS,R3        ;GET NEXT REGISTER FOR BYTE TEST.
013124  005013                  14$:    CLR     (R3)            ;MAKE SURE WORD IS =0.
013126  012705  000010                  MOV     #BIT3,R5        ;SET EXPECTED
013132  110513                  15$:    MOVB    R5,(R3)         ;WRITE BYTE (LOW) BIT3
013134  011304                  16$:    MOV     (R3),R4         ;READ RESULT
013136  020504                  17$:    CMP     R5,R4           ;OK?
013140  001403                  18$:    BEQ     19$             ;
013142  104000                  20$:    HLT     0               ;DVSRS WRONG
013144  012705  000410          19$:    MOV     #BIT8+BIT3,R5   ;SET EXPECTED
013150  110513  000001          21$:    MOVB    #BIT0,1(R3)     ;WRITE BYTE (HIGH) BIT0 (BIT8 OF WORD)
013154  011304                  22$:    MOV     (R3),R4         ;READ RESULT
013156  020504                  23$:    CMP     R5,R4           ;OK
013160  001403                  24$:    BEQ     25$             ;
013162  104000                  26$:    HLT     0               ;DVSRS FAILED
013164  104400                  25$:    SCOPE                   ;SCOPE 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
                                ;--------------
                                ;
013470  012737  000054  001226  TST54:  MOV     #54,TSTNO
013476  012737  013724  001216          MOV     #TST54,NEXT
013504  012737  013524  001220          MOV     #1$,LOOK
013512  005000                          CLR     R0                      ;SEL LINE NUMBER TO ZERO
013514  005001                          CLR     R1                      ;GET SEC. REG TO ZERO ALSO
013516  013702  001370                  MOV     DVSRA,R2                ;SET ADDRESS POINTER INTO R2
013522  110077  160170          1$:     MOVB    R0,@DVSR9               ;LOAD LINE NUMBER
013526  110177  160140                  MOVB    R1,@DVSR5H              ;LOAD SEC. REG.
013532  005004                          CLR     R4                      ;ZERO DATA PATTERN
013534  010412                  2$:     MOV     R4,(R2)                 ;LOAD DATA INTO DV11 REGISTER
013536  011203                          MOV     (R2),R3                 ;READ BACK THE DATA
013540  020403                          CMP     R4,R3                   ;WAS WHAT WAS WRITTEN READ BACK??
013542  001401                          BEQ     64$                     ;BR IF DATA OK
013544  104002                          HLT     2                       ;SECONDARY REGISTER READ WRITE ERROR.
013546  104401                  64$:    SCOP1                           ;LOCK ON DATA,LINE,AND SEC REG? (SW09=1)
013550  005104                          COM     R4                      ;MAKE DATA ALL 1'S
013552  001370                          BNE     2$                      ;RETURN AND WRITE IT INTO THE REGISTER
013554  012704  000001                  MOV     #1,R4                   ;GET READY FOR THE "FLOATING 1"
013560  012737  013570  001220          MOV     #3$,LOOK                ;SET IF SW09=1
013570  010412                  3$:     MOV     R4,(R2)                 ;LOAD DATA
013572  011203                          MOV     (R2),R3                 ;READ DATA
013574  020403                          CMP     R4,R3                   ;DATA OK??
013576  001401                          BEQ     65$                     ;BR IF OK!
013600  104002                          HLT     2                       ;SECONDARY REGISTER READ/WRITE ERROR
013602  104401                  65$:    SCOP1                           ;SW09=1?
013604  000241                          CLC                             ;ZERO CPU CARRY
013606  006104                          ROL     R4                      ;FLOAT THE 1
013610  001365                          BNE     3$                      ;IF NOT DONE WRITE NEW PATTERN
013612  012704  177776                  MOV     #^C(1),R4               ;GET READY FOR THE "FLOATING 0"
013616  012737  013624  001220          MOV     #4$,LOOK                ;SET IF SW09=1
013624  010412                  4$:     MOV     R4,(R2)                 ;WRITE DATA
013626  011203                          MOV     (R2),R3                 ;READ DATA
013630  020403                          CMP     R4,R3                   ;DATA OK??
013632  001401                          BEQ     66$                     ;BR IF OK!
013634  104002                          HLT     2                       ;SECONDARY REGISTER DATA COMPARE ERROR
013636  104401                  66$:    SCOP1                           ;SW09=1??
013640  000261                          SEC                             ;SET CPU CARRY
013642  006104                          ROL     R4                      ;CHANGE DATA PATTERN
013644  103767                          BCS     4$                      ;IF NOT DONE ,GO BACK WITH NEW PATTERN
013646  005012                          CLR     (R2)                    ;ZERO THE REGISTER (ALL DONE WITH THIS ONE.
013650  005201                          INC     R1                      ;UPDATE THE SEC REG POINTER
013652  022701  000020                  CMP     #16.,R1                 ;ALL SEC REGISTERS DONE?
013656  001322                          BNE     1$                      ;BR IF NO.
013660  005001                          CLR     R1                      ;ZERO SEC REG POINTER
013662  005200                          INC     R0                      ;UPDATE LINE NUMBER POINTER
013664  022700  000020                  CMP     #16.,R0                 ;ALL LINES DONE?
013670  001315                          BNE     1$                      ;BR IF NO
```

```
 3017  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  TST55:  MOV     #55,TSTNO
013702  012737  014034  001216         MOV     #TST56,NEXT
013710  012737  013770  001220         MOV     #2$,LOCK
013716  005000                        CLR     R0              ;SELECT THE LINE NUMBER.
013720  005001                  65$:  CLR     R1              ;SET FOR SEC. REG. POINTER.
013722  005004                        CLR     R4              ;SET DATA
013724  013702  001376                MOV     DVSRA,R2        ;SET ACCESS REGISTER.
013730  110077  165436                MOVB    R0,@DVSRS       ;SELECT THE LINE NUMBER
013734  110177  165434          1$:   MOVB    R1,@DVSRSH      ;SELECT THE SEC. REG.
013740  010412                        MOV     R4,(R2)         ;WRITE SEC. REG.
013742  062704  010421                ADD     #^B<0001000100010001>,R4
                                                              ;UPDATE DATA
013746  005201                        INC     R1              ;UPDATE SECONDARY REG. POINTER
013750  022701  000020                CMP     #16.,R1         ;ALL SEC. REG. DONE?
013754  001367                        BNE     1$              ;BR IF NO
013756  005001                        CLR     R1              ;RESET SEC. REG. POINTER TO ZERO.
013760  005004                        CLR     R4              ;ZERO DATA COMPARE
013762  012737  013770  001220         MOV     #2$,LOCK        ;SET FOR LOCK.
013770  110177  165400          2$:   MOVB    R1,@DVSRSH      ;GET SEC. REG.
013774  011203                        MOV     (R2),R3         ;READ SEC. REG.
013776  020403                        CMP     R4,R3           ;R4=GOOD; R3=UNKNOWN
014000  001401                        BEQ     3$              ;BR IF ALL OK
014002  104002                        HLT     2               ;SECONDARY REGISTER ADDRESSING ERROR
014004  104401                  3$:   SCOP1                   ;LOCK ON REG. (SW09=1)
014006  062704  010421                ADD     #^B<0001000100010001>,R4
014012  005201                        INC     R1              ;UPDATE SEC REG POINTER
014014  022701  000020                CMP     #16.,R1         ;ALL 16 LINES TESTED YET?
014020  001363                        BNE     2$              ;BR IF NO
014022  005200                        INC     R0              ;UPDATE LINE NO POINTER
014024  022700  000020                CMP     #16.,R0         ;ALL LINES DONE??
014030  001333                        BNE     65$             ;BR IF NO
014032  104400                        SCOPE                   ;SCOPE THE TEST


                                              :************************ 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
                                          ;--------------
014034  012737  000056  001226  TST56:  MOV     #56,TSTNO
```

```
                014042  012737  000002  001222          MOV     #2,ICOUNT
                014050  012737  014432  001216          MOV     #TST57,NEXT
                014056  012737  014150  001220          MOV     #4$,LOCK
                014064  005000                          CLR     R0              ;SELECT THE LINE NUMBER
                014066  005001                          CLR     R1              ;R1 = SECONDARY REGISTER SELECT.
                014070  005004                          CLR     R4              ;SET DATA TO BE WRITTEN.
                014072  013702  001376                  MOV     DVSRA,R2        ;SET "SECONDARY REGISTER ACCESS" POINTER.
                014076  110077  165270          1$:     MOVB    R0,@DVSRS       ;GET THE LINE NUMBER SELECTED.
                014102  110177  165266          2$:     MOVB    R1,@DVSRSH      ;GET THE SECONDARY REGISTER.
                014106  010412                          MOV     R4,(R2)         ;WRITE THE SECONDARY REG.
                014110  005201                          INC     R1              ;UPDATE  SECONDARY REG. POINTER.
                014112  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. DONE?
                014116  001371                          BNE     2$              ;BR IF NO.
                014120  005001                          CLR     R1              ;ZERO SEC.REG.POINTER
                014122  062704  010421                  ADD     #^B<0001000010000100001>,R4
                                                                                ;UPDATE DATA POINTER.
                014126  005200                          INC     R0              ;UPDATE LINE POINTER.
                014130  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE?
                014134  001360                          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          3$:     MOVB    R0,@DVSRS       ;SELECT LINE NUMBER.
                014150  110177  165220          4$:     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$:     SCOP1                   ;LOCK ON REGISTER? (SW09=1)
                014166  005201                          INC     R1              ;UPDATE SEC. REG. POINTER
                014170  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. FOR THIS LINE DONE?
                014174  001365                          BNE     4$              ;BR IF NO.
                014176  062704  010421                  ADD     #^B<0001000010000100001>,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 DONE FILLING 1'S.
                                        CLR     R0                  ;ZERO LINE # POINTER
                                        CLR     R1                  ;ZERO  SEC REG # POINTER
                        177776  2$:     MOV     #↑C<BIT0>,R4        ;SET INITAL DATA
                        165106  3$:     MOVB    R0,@DVSRS           ;LOAD LINE #
                        165104          MOVB    R1,@DVSRSH          ;LOAD SEC REG #
                                        MOV     R4,(R2)             ;LOAD DATA
                        165074          CLR     @DVSRS              ;ZERO POINTERS
                        177777  100$:   CMP     #↑B<1111111111111111>,(R2)
                                                                    ;VERIFY ONLY ONE LOC. HAS ONE BIT CLEARED
                                        BEQ     5$                  ;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     2                   ;WRONG LINE HAS CLEARED BIT!
                        165044  4$:     CMPB    R1,@DVSRSH          ;IS THIS THE RIGHT SEC REG?
                                        BEQ     5$                  ;BR IF YES
                                        HLT     2                   ;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$:     ADD     #↑C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,@DVSRS
                                        BNE     100$                ;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$:     MOV     R4,@DVSCR       ;LOAD DVSCR WITH EA BITS.
014512  110077  164654                  MOVB    R0,@DVSRS       ;SEL THE LINE NUMBER
014516  117701  164534                  MOVB    @TEMP5,R1       ;GET SEC REG.
014522  110177  164646                  MOVB    R1,@DVSRSH      ;SEL THE SEC. REGISTER
014526  005077  164644                  CLR     @DVSRA          ;HIT THE SEC.REG. ACCESS REGISTER.
014532  017703  164632                  MOV     @DVLCR,R3       ;SAVE THE DVLINE PARM. REG.
014536  042703  177717                  BIC     #^C<BIT5+BIT4>,R3
                                                                ;CLEAR ALL BUT BITS 5 AND 4.
014542  020403                          CMP     R4,R3           ;ARE THE EA BITS GOOD
014544  001401                          BEQ     3$
014546  104004                          HLT     4
014550  104401          3$:             SCOP1                   ;SW09=1?
014552  005237  001256                  INC     TEMP5           ;POP POINTER
014556  005305                          DEC     R5              ;ALL SEC REG DONE?
014560  001352                          BNE     2$              ;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 BIT5,4,1,0)
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
                                ;-------------
D16C40  D16E3E  000061  001E0E  TST61:  MOV     #61,TSTNO
D16C44  D16E42  D16E0A  001E12          MOV     #TST62,NEXT
D16C48  D16E46  000340  177776          MOV     #340,@#PS               ;LOCK OUT INTERUPTS.
D16C4C  D16E4A  001364  001E0E          MOV     DVSCR,R3                ;SET REGISTER POINTER FOR LOADING
D16C50  D16E4E  105013                  CLRB    @#DVSRA                 ;CLEAR LINE POINTER
D16C52  D16E50  105013                  CLRB    @#DVSRA                 ;CLEAR ACCESS REG.
D16C54  D16E52  012723  000004          MOV     #4(BIT11),(R3)+
D16C58  D16E56  012723  177777          MOV     #-1,(R3)+               ;SET ALL BITS BUT MSTCLR
D16C5C  D16E5A  005723          10:     TST     -(R3)                   ;LOAD ALL OTHER REGISTERS WITH ALL 1'S
D16C5E  D16E5C  100375                  BPL     10                      ;ALLREGISTERS LOADED?
D16C60  D16E5E  005003                  CLR     R3                      ;BR IF NO
D16C62  D16E60  012703  004000          MOV     #MRESET,@DVSCR          ;ISSUE A "MASTER CLEAR"
D16C66  D16E64  001364                  MOV     DVSCR,R3                ;SET REGISTER POINTER
D16C68  D16E66  005000                  CLR     R0                      ;SET "EXPECTED" FOR DVSCR
D16C6A  D16E68  011304                  MOV     (R3),R4                 ;READ THE DVSCR REG.
D16C6C  D16E6A  020004                  CMP     R0,R4                   ;IS BITS ALONE SET?
D16C6E  D16E6C  001403                  BEQ     20                      ;BR IF YES
D16C70  D16E6E  104000                  TRAP    0                       ;DVSCR HAS WRONG DATA)
D16C72  D16E70  005723          20:     TST     (R3)+                   ;POP POINTER TO DVRIC
D16C74  D16E72  005000                  CLR     R0                      ;SET EXPECTED TO ZERO
D16C76  D16E74  011304                  MOV     (R3),R4                 ;DVRIC (EXPECT ALL 0'S)
D16C78  D16E76  001403                  BEQ     30                      ;BR IF OK
D16C7A  D16E78  104000                  TRAP    0                       ;DVRIC NO ALL 0'S
D16C7C  D16E7A  005723          30:     TST     (R3)+                   ;POP POINTER TO DVLCR REG
D16C7E  D16E7C  011304                  MOV     (R3),R4                 ;DVLCR (READ DVLCR INTO R4)
D16C80  D16E7E  042704  000063          BIC     #BIT5+BIT4+BIT1+BIT0,R4
D16C84  D16E82  020004                  CMP     R0,R4                   ;DISREGUARD BR TEST POINTS AND MEM EXT BITS.
D16C86  D16E84  001403                  BEQ     40                      ;DVLCR OK?
D16C88  D16E86  104000                  TRAP    0                       ;DVLCR INCORECT (DISREGUARD BITS,4,1,0)
D16C8A  D16E88  005723          40:     TST     (R3)+                   ;POP POINTER TO DVSRS REG
D16C8C  D16E8A  011304                  MOV     (R3),R4                 ;DVSRS (EXPECT ALL 0'S)
D16C8E  D16E8C  001403                  BEQ     50                      ;BR IF OK
D16C90  D16E8E  104000                  TRAP    0                       ;DVSRS REG NOT ALL ZEROS
D16C92  D16E90  005723          50:     TST     (R3)+                   ;POP POINTER TO DVSRA REG
D16C94  D16E92  011304                  MOV     (R3),R4                 ;DVSRA (EXPECT ALL 0'S)
D16C96  D16E94  001403                  BEQ     60                      ;BR IF GOOD
D16C98  D16E96  104000                  TRAP    0                       ;DVSRA NOT ALL 0'S
D16C9A  D16E98  005723          60:     TST     (R3)+                   ;POP POINTER TO DVSFR
D16C9C  D16E9A  011304                  MOV     (R3),R4                 ;DVSFR (EXPECT ALL 1'S (THATS RIGHT))
D16C9E  D16E9C  012705  177777          MOV     #177777,R5              ;SET EXPECTED
D16CA2  D16EA0  020504                  CMP     R5,R4                   ;EXPECETD =FOUND?
D16CA4  D16EA2  001403                  BEQ     70                      ;BR IF YES
D16CA6  D16EA4  104000                  TRAP    0                       ;DVSFR NOT ALL 1'S
D16CA8  D16EA6  005723          70:     TST     (R3)+                   ;POP POINTER TO DVMSR REG
D16CAA  D16EA8  005713                  TST     (R3)                    ;DVMSR S/B PLUS (15=0)
D16CAC  D16EAA  100001                  BPL     80#
```

```
                                                HLT     0
                                        64$:    TST     (R3)+           ;POP POINTER TO RESV16 REG
                                                MOV     (R3),R4         ;RESV16 (EXPECT ALL 0'S)
                                                CLR     R3              ;SET EXPECTED TO 0'S
                                                CMP     R3,R4           ;WELL, DOES IT =1'S?
                                                BEQ     66$             ;BR IF OK
                                                HLT     0               ;RESV16 NOT ALL 0'S
                                        66$:    SCOPE                   ;SCOPE THIS TEST:


                                        ;+++++++++++++++++++++++++ TEST 62 ++++++++++++++++++++++++++++++
                                        ;+ATTACK OF THE SPECIAL FUNCTIONS REGISTER.
                                        ;+BEGIN CHECK OF THE DVSFR.
                                        ;+SUMMARY OF PROC. INSTRUCTIONS
                                        ;+
                                        ;+BIT14 BIT13 BIT12  INSTRUCTION
                                        ;+  0     0     0    BRANCH "A"
                                        ;+  0     0     1    ALU OPERATION
                                        ;+  0     1     0    ARM OPERATION
                                        ;+  0     1     1    DATA TRANSFER
                                        ;+  1     0     0    NPR OPERATION
                                        ;+  1     0     1    SET/CLEAR OPERATION
                                        ;+  1     1     0    BCC CALCULATION
                                        ;+  1     1     1    BRANCH "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
                                        ;--------------
                                        TST62:  MOV     #62,ISTNO
                                                MOV     #TST63,NEXT
                                                MSTCLR                  ;CLEAR ALL THE DV11
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                MOV     #6C.C,RS        ;PUT INSTR INTO DVSFR
                                                MOV     RS,@DVSFR       ;
                                                CMP     @DVSFR,RS       ;WAS THE DVSFR REALLY LOADED?
                                                BEQ     1$              ;BR IF YES
                                                HLT     1               ;BAD DVSFR
                                        1$:     BIC     #BIT3,@DVSCR    ;CLEAR SOURCE SEL.
                                                ROMCLK                  ;ISSUE A ROM CLOCK
                                                NOP                     ;WAIST AN INTRUSTION TIME
                                                BIT     #BIT1,@DVSCR    ;DID CLK BIT CLEAR BY IT SELF?
                                                BEQ     2$              ;BR IF CLK GONE
                                                HLT     2               ;BIT 1 OF DVSCR (ROM CLK) NOT ZERO
                                        2$:     CMP     R5,@DVSFR       ;HAS DATA IN DVSFR CHANGED?
                                                BNE     3$              ;BR IF YES
                                                HLT     3               ;DATA NOT CHANGED (DID CLK REALLY CLR???)
                                        3$:     SCOPE                   ;SCOPE THIS TEST.
```

```
                                        ;********************** TEST 63 ******************************
                                        ;*BASIC TEST OF DVSFR
                                        ;*TEST THAT "BRANCH A" INSTRUCTION.
                                        ;*POINTS TESTED:
                                        ;*
                                        ;*BIT11  BIT10  BIT09  BIT08  BR "A"  BR "B"  FUNCTION
                                        ;*  0      0      0      1      L.H    H.H    PLUS 3 VOLTS
                                        ;*  0      0      1      1      L.H    H.H    DVSCR08 (RQ+1)
                                        ;*  0      1      0      1      H.H    H.H    NPR SILO NOT AVAIL.
                                        ;*  0      1      1      1      H.H    H.H    SILO FULL
                                        ;*  0      1      1      0      H.H    H.H    NXM
                                        ;*
                                        ;*************************************************************

                                ;  TEST 63
                                ;--------------
016000  016737  000063  001606  TST63:  MOV     #63,TSTNO
016006  016700  016626  001216          MOV     #TST64,NEXT
016014  104416                          MTCLR                   ;CLEAR DV11
016016  016700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER TO R0
016022  016701  001370                  MOV     DVLCR,R1        ;SET DVLCR POINTER TO R1
016026  012777  000410  163744          MOV     #BIT8+BIT3,@DVSCR
016034  012710  000400          10:     MOV     #BIT8,(R0)      ;BR-A TEST +3
016040  011002                  MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016042  012705  000002                  MOV     #BIT1,R5        ;SET EXPECTED RESULTS
016046  011304                  MOV     (R3),R4         ;READ DVLCR INTO R4
016050  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                        ;CLEAR UNWANTED BITS.
016054  020504                  CMP     R5,R4           ;EXPECTED = FOUND??
016056  001401                  BEQ     20$             ;BR IF OK
016060  104006                  TRAP    6               ;BR POINT WRONG
016062  012710  002400          30:     MOV     #BIT10+BIT9,(R0)
016066  011002                  MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016070  012705  000002                  MOV     #BIT1,R5        ;SET EXPECTED RESULTS
016074  011304                  MOV     (R3),R4         ;READ DVLCR INTO R4
016076  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                        ;CLEAR UNWANTED BITS.
016102  020504                  CMP     R5,R4           ;EXPECTED = FOUND??
016104  001401                  BEQ     30$             ;BR IF OK
016106  104006                  TRAP    6               ;BR POINT WRONG
016110  012777  000400  163662  38:     MOV     #BIT8,@DVSCR    ;RESET BIT 8 TO =0
016116  011002                  MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016120  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016124  011304                  MOV     (R3),R4         ;READ DVLCR INTO R4
016126  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                        ;CLEAR UNWANTED BITS.
016132  020504                  CMP     R5,R4           ;EXPECTED = FOUND??
016134  001401                  BEQ     40$             ;BR IF OK
016136  104006                  TRAP    6               ;BR POINT WRONG
016140  012710  003400          48:     MOV     #BIT10+BIT9+BIT8,(R0)
016144  011002                  MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016146  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016152  011304                  MOV     (R3),R4         ;READ DVLCR INTO R4
016154  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                        ;CLEAR UNWANTED BITS.
016160  020504                  CMP     R5,R4           ;EXPECTED = FOUND??
```

```
016544  001401                  BEQ     5$              ;BR IF Ok
016546  104006                  HLT     6               ;BR POINT WRONG
016550  012710  007400      5$: MOV     #BIT11+BIT10+BIT9+BIT8,(R0)
016554  011002                  MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016556  012705  000003          MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016562  011304                  MOV     (R3),R4         ;READ DVLCR INTO R4
016564  042704  177774          BIC     #^C<BIT1+BIT0>,R4
                                                        ;CLEAR UNWANTED BITS.
016570  020504                  CMP     R5,R4           ;EXPECTED = FOUND??
016572  001401                  BEQ     6$              ;BR IF OK
016574  104006                  HLT     6               ;BR POINT WRONG
016576  012710  003000      6$: MOV     #BIT10+BIT9,(R0)        ;NXM
016602  011002                  MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016604  012705  000003          MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016610  011304                  MOV     (R3),R4         ;READ DVLCR INTO R4
016612  042704  177774          BIC     #^C<BIT1+BIT0>,R4
                                                        ;CLEAR UNWANTED BITS.
016616  020504                  CMP     R5,R4           ;EXPECTED = FOUND??
016620  001401                  BEQ     7$              ;BR IF OK
016622  104006                  HLT     6               ;BR POINT WRONG
016624  104400                  SCOPE                   ;SCOPE TEST
```

```
                        ;********************** TEST 64 ****************************
                        ;*TEST OF BRANCH 3"
                        ;*TEST THAT POINT 16 (GROUND)
                        ;*MAKES LCR BIT1=1 AND BIT0=1.
                        ;;*********************************************************

                        ;  TEST 64
                        ;--------------
016626  016737  000064  001336  TST64:  MOV     #64,TSTNO
016634  012737  015704  001216          MOV     #TST65,NEXT
016642  104412                  MSTCLR                  ;CLEAR DV11
016644  012777  000010  163510          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
016652  012777  077400  163520          MOV     #BRB+BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                        ;BR-B "GROUND"?
016660  017702  163514                  MOV     @DVSFR,R2       ;READ DVSFR INTO R2
016664  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016670  017704  163474                  MOV     @DVLCR,R4       ;READ REAL RESULTS
016674  020504                  CMP     R5,R4           ;SAME??
016676  001401                  BEQ     1$              ;
016700  104006                  HLT     6               ;BR TEST POINT WRONG
016702  104400                  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 BUS
                        ;*   1     0       1       0     MEMORY PARITY ERROR
                        ;*   1     1       1       1     WRITE INHIBIT
                        ;;*********************************************************

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

```
                016704  016737  000065  001226  TST65:  MOV     #65,ISTNO
                016712  016737  016072  001216          MOV     #TST66,NEXT
                016720  104414                          MSTCLR
                016722  013700  001400                  MOV     DVSFR,R0
                016726  013703  001370                  MOV     DVLCR,R3
                016732  012777  000010  163-33          MOV     #BIT3,@DVSCR
                016740  012710  074000          1$:     MOV     #GRB+BIT11,(R0)
                016744  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                016746  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                016752  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
                016754  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                016760  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
                016762  001401                          BEQ     2$              ;BR IF OK
                016764  104006                          HLT     6               ;BR POINT WRONG
                016766  012710  074400          2$:     MOV     #GRB+BIT11+BIT8,(R0)
                016772  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                016774  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                017000  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
                017002  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                017006  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
                017010  001401                          BEQ     3$              ;BR IF OK
                017012  104006                          HLT     6               ;BR POINT WRONG
                017014  012710  075000          3$:     MOV     #GRB+BIT11+BIT9,(R0)
                017020  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                017022  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                017026  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
                017030  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                017034  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
                017036  001401                          BEQ     4$              ;BR IF OK
                017040  104006                          HLT     6               ;BR POINT WRONG
                017042  012710  077000          4$:     MOV     #GRB+BIT11+BIT10+BIT9,(R0)
                017046  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
                017050  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
                017054  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
                017056  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                                ;CLEAR UNWANTED BITS.
                017062  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
                017064  001401                          BEQ     5$              ;BR IF OK
                017066  104006                          HLT     6               ;BR POINT WRONG
                017070  104400          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
```

```
                                        ;--------------
016072  012737  000066  001226  TST66:  MOV     #66,TSTNO
016100  012737  016664  001216          MOV     #TST67,NEXT
016106  012737  016136  001220          MOV     #1$,LOCK
016114  104412                          MSTCLR                  ;CLEAR DV11
016116  052777  000400  163336          BIS     #BIT8,@DVSCR    ;SET BIT8.
016124  052777  000010  163330          BIS     #BIT3,@DVSCR    ;SET SOURCE SEL.
016132  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER ADDRESS IN R0
016136  012760  050016  000000  1$:     MOV     #S.C+BIT3+BIT2+BIT1,(R0)
016144  052760  000010  000000          MOV     #BIT3,R5        ;DO SET/CLEAR -CLEAR BIT 8 OF DVSCR
016152  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
016154  104415                          ROMCLK                  ;CYCLE THE ROM
016156  017704  163304                  MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
016162  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
016164  001403                          BEQ     64$             ;BR IF DVSCR OK
016166  104006                          HLT     6               ;ROM FAILED TO EXECUTE
016170  104401                  64$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
016172  012737  016174  001220          MOV     #3$,LOCK        ;SET FOR RETURN IF SW09=1
016200  052705  000200          3$:     BIS     #BIT7,R5        ;SET EXPECTED (SCR BIT 7=1)
016204  012760  050013  000000          MOV     #S.C+BIT3+BIT1+BIT0,(R0)
016212  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
016214  104415                          ROMCLK                  ;CYCLE THE ROM
016216  017704  163146                  MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
016222  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
016224  001403                          BEQ     65$             ;BR IF DVSCR OK
016226  104006                          HLT     6               ;ROM FAILED TO EXECUTE
016230  104401                  65$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
016232  012737  016234  001220          MOV     #4$,LOCK
016240  052705  002000          4$:     BIS     #BIT10,R5       ;ALTER EXPECTED ADDRESS
016244  012760  050013  000000          MOV     #S.C+BIT3+BIT1,(R0)
                                                                ;DO A SET/CLEAR SET DVSCR BIT 10
016252  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
016254  104415                          ROMCLK                  ;CYCLE THE ROM
016256  017704  163110                  MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
016262  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
016264  001403                          BEQ     66$             ;BR IF DVSCR OK
016266  104006                          HLT     6               ;ROM FAILED TO EXECUTE
016270  104401                  66$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
016272  104420                          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      0     SET BIT# 15
                                        ;*   1      1      1      0      0      1     SET BIT# 14
                                        ;*   1      1      1      0      1      0     SET BIT# 13
                                        ;*   1      1      1      1      0      1     SET BIT# 12
                                        ;*
                                        ;;*************************************************************

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

```
016624  016737  000267  001226  TST67:  MOV     #67,TSTNO
016632  016737  016606  001216          MOV     #TST70,NEXT
016640  012736  177060                  MOV     #177060,@#DVSFR         ;SET DVSFR TO 0
016646  104415                          MSTCLR                          ;CLEAR DV11
016650  012737  001400  177060          MOV     #BIT8,@#DVSFR           ;SET DVSFR TO RO
016656  012737  000010  163046          MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
016664  012705  177777                  MOV     #^B1111111111111111,R5  ;SET EXPECTED RESULTS
016670  012710  000010                  MOV     #15.-BIT3,(R0)
                                                                        ;SET/CLEAR DVRICR 15
016674  011002                          MOV     (R0),R2                 ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
016676  104415                          ROMCLK                          ;CYCLE THE ROM
016700  017704  163032                  MOV     @DVRIC,R4               ;READ DVRIC INTO "FOUND" LOC.
016704  020504                          CMP     R5,R4                   ;WAS THE ROM INSTR EXECUTED?
016706  001401                          BEQ     64$                     ;BR IF DVSCR OK
016710  000000                          HALT                            ;ROM FAILED TO EXECUTE
016712  104401                  64$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
016714  104415                          MSTCLR                          ;CLEAR DV11
016716  012737  000010  163006          MOV     #BIT3,@#DVSCR           ;SET SOURCE SEL
016724  012705  040000                  MOV     #^B0100000000000000,R5  ;SET EXPECTED RESULTS
016730  012710  000011                  MOV     #15.-BIT3-BIT0,(R0)
                                                                        ;SET/CLEAR DVRICR 14
016734  011002                          MOV     (R0),R2                 ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
016736  104415                          ROMCLK                          ;CYCLE THE ROM
016740  017704  162772                  MOV     @DVRIC,R4               ;READ DVRIC INTO "FOUND" LOC.
016744  020504                          CMP     R5,R4                   ;WAS THE ROM INSTR EXECUTED?
016746  001401                          BEQ     65$                     ;BR IF DVSCR OK
016750  000000                          HALT                            ;ROM FAILED TO EXECUTE
016752  104401                  65$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
016754  104415                          MSTCLR                          ;CLEAR DV11
016756  012737  000010  162746          MOV     #BIT3,@#DVSCR           ;SET SOURCE SEL
016764  012705  020000                  MOV     #^B0010000000000000,R5  ;SET EXPECTED RESULTS
016770  012710  000012                  MOV     #15.-BIT3-BIT2,(R0)
                                                                        ;SET/CLEAR DVRICR 13
016774  011002                          MOV     (R0),R2                 ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
016776  104415                          ROMCLK                          ;CYCLE THE ROM
017000  017704  162732                  MOV     @DVRIC,R4               ;READ DVRIC INTO "FOUND" LOC.
017004  020504                          CMP     R5,R4                   ;WAS THE ROM INSTR EXECUTED?
017006  001401                          BEQ     66$                     ;BR IF DVSCR OK
017010  000000                          HALT                            ;ROM FAILED TO EXECUTE
017012  104401                  66$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
017014  104415                          MSTCLR                          ;CLEAR DV11
017016  012737  000010  162706          MOV     #BIT3,@#DVSCR           ;SET SOURCE SEL
017024  012705  010000                  MOV     #^B0001000000000000,R5  ;SET EXPECTED RESULTS
017030  012710  000013                  MOV     #15.-BIT3-BIT2-BIT0,(R0)
                                                                        ;SET/CLEAR DVRICR 12
017034  011002                          MOV     (R0),R2                 ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
017036  104415                          ROMCLK                          ;CYCLE THE ROM
017040  017704  162672                  MOV     @DVRIC,R4               ;READ DVRIC INTO "FOUND" LOC.
017044  020504                          CMP     R5,R4                   ;WAS THE ROM INSTR EXECUTED?
017046  001401                          BEQ     67$                     ;BR IF DVSCR OK
017050  000000                          HALT                            ;ROM FAILED TO EXECUTE
017052  104401                  67$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
017054  104403                          SCOPE                           ;SCOPE THIS TEST


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

```
                                        ;*TEST OF "SET/CLEAR" AND
                                        ;*"BRANCH A" AND "BRANCH B" FUNCTIONS.
                                        ;;*****************************************************************

                                ;  TEST 70
                                ;--------------
016506  012737  000070  001226  TST70:  MOV     #70,ISTNO
016512  012737  017170  001216          MOV     #TST71,NEXT
016520  012737  016544  001220          MOV     #1$,LOCK
016526  104412                          MSTCLR                  ;CLEAR DV11
016530  012737  000010  162622          MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
016540  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER INTO R0

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR ALU BIT03
016544  004237  017134          1$:     JSR     R2,10$          ;GOTO SUBROUTINE.
016550  000054                          BIT4+BIT2               ;POINT SET/CLEARED
016552  006004                          BIT11+BIT10             ;BR TEST POINT
016554  000002                          BIT1                    ;EXPECTED RESULTS IN DVLCR
016556  004237  017134                  JSR     R2,10$          ;GOSUB
016562  000055                          BIT4+BIT2+BIT0          ;POINT SET/CLEARED
016564  006004                          BIT11+BIT10             ;BR TEST POINT
016566  000003                          BIT1+BIT0               ;EXPECTED RESUTS IN DVLCR
016570  104401                          SCOP1                   ;SWR09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT00
016572  012737  016600  001220          MOV     #2$,LOCK        ;SET RETURN IF SW09=1
016600  004237  017134          2$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016604  000047                          BIT5+BIT2+BIT1+BIT0             ;POINT SET/CLEARED [SET]
016606  070000                          BRB                     ;BR TEST POINT
016610  000001                          BIT0                    ;DVLCR EXPECTED
016612  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016616  000043                          BIT5+BIT1+BIT0                  ;POINT SET/CLEARED [CLEARED]
016620  070000                          BRB                     ;BR TEST POINT
016622  000003                          BIT1+BIT0               ;EXPECTED RESULTS
016624  104401                  65$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT01
016626  012737  016634  001220          MOV     #3$,LOCK        ;SET RETURN IF SW09=1
016634  004237  017134          3$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016640  000045                          BIT5+BIT2+BIT0                  ;POINT SET/CLEARED [SET]
016642  070400                          BRB+BIT8                        ;BR TEST POINT
016644  000001                          BIT0                    ;DVLCR EXPECTED
016646  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016652  000041                          BIT5+BIT0                       ;POINT SET/CLEARED [CLEARED]
016654  070400                          BRB+BIT8                        ;BR TEST POINT
016656  000003                          BIT1+BIT0               ;EXPECTED RESULTS
016660  104401                  67$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT02
016662  012737  016670  001220          MOV     #4$,LOCK        ;SET RETURN IF SW09=1
016670  004237  017134          4$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016674  000046                          BIT5+BIT2+BIT1                  ;POINT SET/CLEARED [SET]
```

```
016676  071000                          BNB+BIT9                        ;BR TEST POINT
016700  000001                          BIT0                    ;DVLCR EXPECTED
016702  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016706  000042                          BIT5+BIT1                       ;POINT SET/CLEARED [CLEARED]
016710  071000                          BRB+BIT9                        ;BR TEST POINT
016712  000003                          BIT1+BIT0               ;EXPECTED RESULTS
016714  104401                  69$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT03
016716  012737  016724  001220          MOV     #6$,LOCK                ;SET RETURN IF SW09=1
016724  004237  017134          6$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016730  000044                          BIT5+BIT2                       ;POINT SET/CLEARED [SET]
016732  071400                          BRB+BIT9+BIT8                   ;BR TEST POINT
016734  000001                          BIT0                    ;DVLCR EXPECTED
016736  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016742  000040                          BIT5                    ;POINT SET/CLEARED [CLEARED]
016744  071400                          BRB+BIT9+BIT8                   ;BR TEST POINT
016746  000003                          BIT1+BIT0               ;EXPECTED RESULTS
016750  104401                  71$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT04
016752  012737  016760  001220          MOV     #7$,LOCK                ;SET RETURN IF SW09=1
016760  004237  017134          7$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016764  000207                          BIT7+BIT2+BIT1+BIT0                     ;POINT SET/CLEARED [SET]
016766  072000                          BRB+BIT10                       ;BR TEST POINT
016770  000001                          BIT0                    ;DVLCR EXPECTED
016772  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
016776  000203                          BIT7+BIT1+BIT0                  ;POINT SET/CLEARED [CLEARED]
017000  072000                          BRB+BIT10                       ;BR TEST POINT
017002  000003                          BIT1+BIT0               ;EXPECTED RESULTS
017004  104401                  73$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT05
017006  012737  017014  001220          MOV     #8$,LOCK                ;SET RETURN IF SW09=1
017014  004237  017134          8$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
017020  000205                          BIT7+BIT2+BIT0                  ;POINT SET/CLEARED [SET]
017022  072400                          BRB+BIT10+BIT8                  ;BR TEST POINT
017024  000001                          BIT0                    ;DVLCR EXPECTED
017026  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
017032  000201                          BIT7+BIT0                       ;POINT SET/CLEARED [CLEARED]
017034  072400                          BRB+BIT10+BIT8                  ;BR TEST POINT
017036  000003                          BIT1+BIT0               ;EXPECTED RESULTS
017040  104401                  75$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT06
017042  012737  017050  001220          MOV     #9$,LOCK                ;SET RETURN IF SW09=1
017050  004237  017134          9$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
017054  000206                          BIT7+BIT2+BIT1                  ;POINT SET/CLEARED [SET]
017056  073000                          BRB+BIT10+BIT9                  ;BR TEST POINT
017060  000001                          BIT0                    ;DVLCR EXPECTED
017062  004237  017134                  JSR     R2,10$          ;GOTO SUB ROUTINE
017066  000202                          BIT7+BIT1                       ;POINT SET/CLEARED [CLEARED]
```

```
                017070  073000                          BRB+BIT10+BIT9                  ;BR TEST POINT
                017072  000003                          BIT1+BIT0                       ;EXPECTED RESULTS
                017074  104401              77$:        SCOP1                           ;SWR 09=1?

                                                        :*TEST SET/CLEAR FUNCTION
                                                        :*FOR RAM OUTPUT BIT07
                017076  012737  017104  001220        MOV     #101$,LOCK      ;SET RETURN IF SWD9=1
                017104  004237  017134        101$:   JSR     R2,10$  ;GOTO THE SUBROUTINE
                017110  000204                          BIT7+BIT2                       ;POINT SET/CLEARED [SET]
                017112  073400                          BRB+BIT10+BIT9+BIT8                     ;BR TEST POINT
                017114  000001                          BIT0                            ;DVLCR EXPECTED
                017116  004237  017134                  JSR     R2,10$                  ;GOTO SUB ROUTINE
                017122  000200                          BIT7                            ;POINT SET/CLEARED [CLEARED]
                017124  073400                          BRB+BIT10+BIT9+BIT8                     ;BR TEST POINT
                017126  000003                          BIT1+BIT0                       ;EXPECTED RESULTS
                017130  104401              79$:        SCOP1                           ;SWR 09=1?
                017132  104400                          SCOPE                           ;SCOPE THE TEST
                017134  012710  050000        10$:      MOV     #S.C,(R0)               ;SET/CLEAR INSTR
                017140  010201                          MOV     R2,R1                   ;SAVE JSR PC ADDRESS
                017142  052210                          BIS     (R2)+,(R0)              ;LOAD POINT SET/CLEARED
                017144  104415                          ROMCLK                          ;CYCLE THE ROM
                017146  012210                          MOV     (R2)+,(R0)              ;LOAD BR TEST POINT
                017150  011003                          MOV     (R0),R3                 ;READ DVSFR INTO R3
                017152  012205                          MOV     (R2)+,R5                ;LOAD EXPECTED INTO R5
                017154  017704  162210                  MOV     @DVLCR,R4               ;READ DVLCR INTO FOUND LOC.
                017160  020504                          CMP     R5,R4                   ;EXPECTED=FOUND?
                017162  001401                          BEQ     11$                     ;BR IF YES
                017164  104005                          HLT     5                       ;DVLCR WRONG BR RESULTS.
                017166  000202              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  1$:     MOV     #S.C+BIT4+BIT0,@DVSFR
                017232  104415                          ROMCLK                          ;S/C "SILO IN"
                017234  012777  007400  162136          MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                                                        ;BR-A "SILO FULL"?
                017242  017702  162132                  MOV     @DVSFR,R2               ;SAVE CONTENTS OF DVSFR FOR ERROR PRINTOUT
                017246  017704  162116                  MOV     @DVLCR,R4               ;READ DVLCR FOR RESULTS
                017252  020504                          CMP     R5,R4                   ;ARE BR TEST POINTS CORRECT?
                017254  001401                          BEQ     64$                     ;BR IF YES
                017256  104006                          HLT     6                       ;BR TEST POINTS WRONG (BIT1 OR 0)
```

```
 3857  017260  005300                    64$:    DEC     R0              ;IS SILO FULL-1 YET?
 3858  017262  001360                            BNE     1$              ;BR IF NOT 127 TIMES YET
 3859  017264  012777  050021  162106    2$:     MOV     #S.C+BIT4+BIT0,@DVSFR
 3860                                              ;S/C "SILO IN"
 3861  017272  104415                            ROMCLK                  ;
 3862  017274  000240                            NOP                     ;WAIST INSTRUCTION TIME
 3863  017276  012777  007400  162074            MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
 3864                                              ;BR-A "SILO FULL"?
 3865  017304  017702  162070                    MOV     @DVSFR,R2       ;SAVE DVSFR
 3866  017310  017704  162054                    MOV     @DVLCR,R4       ;READ BR TEST POINTS
 3867  017314  042705  000001                    BIC     #BIT0,R5        ;ALTER EXPECTED RESULTS
 3868  017320  020504                            CMP     R5,R4           ;BR TEST POINTS OK??
 3869  017322  001401                            BEQ     3$              ;BR IF YES
 3870  017324  104006                            HLT     6               ;BR TEST POINTS WRONG
 3871  017326  012777  050020  162044    3$:     MOV     #S.C+BIT4,@DVSFR
 3872  017334  104415                            ROMCLK                  ;S/C "SILO OUT"
 3873  017336  000240                            NOP                     ;WAIST INSTR TIME
 3874  017340  012777  007400  162032            MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
 3875                                              ;BR-A "SILO FULL"?
 3876  017346  005002                            CLR     R2              ;DEALAY AT LEAST 32US
 3877  017350  032777  000001  162012    4$:     BIT     #BIT0,@DVLCR    ;IS SILO *NOT FULL*??
 3878  017356  001003                            BNE     5$              ;BR IF OK.
 3879  017360  062702  000001                    ADD     #1,R2           ;DELAY.....
 3880  017364  001371                            BNE     4$              ;GOTO 4$
 3881  017366  017702  162006            5$:     MOV     @DVSFR,R2       ;SAVE DVSFR
 3882  017372  017704  161772                    MOV     @DVLCR,R4       ;READ BR TEST POINTS
 3883  017376  052705  000001                    BIS     #BIT0,R5        ;SET EXPECTED RESULTS
 3884  017402  020504                            CMP     R5,R4           ;OK??
 3885  017404  001401                            BEQ     6$              ;YES
 3886  017406  104006                            HLT     6               ;SILO  STILL FULL.
 3887  017410  104400                    6$:     SCOPE                   ;SCOPE TEST
 3888
 3889                                              ;************************ TEST 72 *****************************
 3890                                              ;*TEST THAT AFTER AN INIT
 3891                                              ;*THAT "RECV CHARACTER WAITING"
 3892                                              ;*IS FALSE (HIGH) AND THEN VERIFY
 3893                                              ;*THAT WHEN "SILO IN" IS ASSERTED THAT
 3894                                              ;*THAT "RCVED CHARACTER WAITING" IS TRUE (LOW)
 3895                                              ;*AND MAKES "BRANCH A" TRUE.
 3896                                              ;:*************************************************************
 3897
 3898                                              ;  TEST 72
 3899                                              ;--------------
 3900  017412  012737  000072  001226    TST72:  MOV     #72,TSTNO
 3901  017420  012737  017616  001216            MOV     #TST73,NEXT
 3902  017426  104412                            MSTCLR                  ;CLEAR DV11
 3903  017430  012777  000010  161724            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 3904  017436  012705  000003                    MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
 3905  017442  012702  001400                    MOV     #BIT9+BIT8,R2   ;BR-A "RCVD CHAR WAITING"?
 3906  017446  010277  161726                    MOV     R2,@DVSFR       ;LOAD DV INSTR
 3907  017452  017704  161712                    MOV     @DVLCR,R4       ;READ TEST POINTS
 3908  017456  020504                            CMP     R5,R4           ;OK??
 3909  017460  001401                            BEQ     64$             ;YES
 3910  017462  104006                            HLT     6               ;TEST POINT RECV CHAR WAITING WRONG
 3911  017464  012702  050021            64$:    MOV     #S.C+BIT4+BIT0,R2
 3912                                              ;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$:         BIC     #BIT0,R5        ;RESET EXPECTED RESULTS
3940  017574  042777  000200  16                    BIC     #BIT7,@DVSCR    ;CLEAR RECV INT
3941  017602  017704  161562                        MOV     @DVLCR,R4       ;READ BR POINTS
3942  017606  020504                                CMP     R5,R4           ;
3943  017610  001401                                BEQ     5$              ;
3944  017612  104006                                HLT     6               ;BR TEST POINTS WRONG
3945  017614  104400            5$:                 SCOPE                   ;SCOPE THIS TEST
3946
3947
3948                                                ;************************ TEST 73 *******************************
3949                                                ;*BASIC TEST OF THE "DATA TRANSFER INSTRUCTION"
3950                                                ;*BITS 07,06,05,04 OF DVSFR INDICATE THE SOURCE
3951                                                ;*BITS 03,02,01,00 OF DVSFR INDICATE THE DESTINATION.
3952                                                ;:************************************************************
3953
3954                                                ;  TEST 73
3955                                                ;--------------
3956  017616  012737  000073  001226  TST73:       MOV     #73,TSTNO
3957  017624  012737  020156  001216                MOV     #TST74,NEXT
3958  017632  012737  017732  001220                MOV     #1$,LOCK
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=CDC 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  000006                         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$:           ADD     #BIT4,R3                ;UPDATE SOURCE REGISTER
017716  022703  000300                         CMP     #300,R3                 ;DON'T DO SILO REGISTER!!
017722  001773                                 BEQ     66$                     ;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  000266                         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.
```

```
                        MOV     #4$,LOCK        ;SET RETURN IF SWO9=1
                4$:     MOV     #BIT4,R5        ;SET EXPECTED DATA
                        MOV     #XFR,R3 ;-DATA XFER-
                        BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
                        JSR     R2,10$          ;S= RAM OUTPUT. D=DVRIC
                        BIT7+BIT2+BIT1+BIT0
                        BIT7+BIT1+BIT0


                        ;+TEST TO SET RAM OUTPUT DATA BIT7
                        ;+AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                        ;+TO PLACE BIT7 INTO THE DVRIC REGISTER.
                        MOV     #6$,LOCK        ;SET RETURN IF SWO9=1
                6$:     MOV     #BIT7,R5        ;SET EXPECTED DATA
                        MOV     #XFR,R3 ;-DATA XFER-
                        BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
                        JSR     R2,10$          ;S= RAM OUTPUT. D=DVRIC
                        BIT7+BIT2
                        BIT7
                        SCOPE
                10$:    MOV     #5.0,(R0) ;SET CLEAR INSTR
                        MOV     R0,R1           ;SAVE PC TO R1
                        BIS     (R2)+,(R0)      ;LOAD SET/CLEAR POINT
                        ROMCLK                  ;EXECUTE
                        MOV     R3,(R0)         ;LOAD XFER
                        ROMCLK                  ;EXECUTE
                        MOV     @DVRIC,R4       ;READ RESULTS
                        CMP     R5,R4           ;OK??
                        BEQ     11$             ;YES
                        HLT     5               ;XFER FAILED
                11$:    MOV     #5.0,(R0) ;SET/CLEAR
                        BIS     (R2)+,(R0)      ;POINT
                        ROMCLK                  ;EXECUTE
                        MOV     R3,(R0)         ;XFER
                        ROMCLK                  ;EXECUTE
                        CLR     R5              ;SET EXPECTED RESULTS
                        MOV     @DVRIC,R4       ;READ REAL RESULTS
                        BEQ     12$             ;BR IF RESULT=0
                        HLT     6               ;DVRIC NOT =0
                12$:    RTS     R2              ;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 DVRIC 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
```

```
                                        ;--------------
000000  000000  012737  000074  001206  TST74:  MOV     #74,ISTNO
000006  000006  012737  000276  001216           MOV     #TST75,NEXT
000014  000014  105746                           MSTCLR                  ;RESET DV11
000016  000016  012777  000010  161160           MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
000024  000024  012700  001400                   MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
000030  000030  012702  030366                   MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1,R2
000034  000034  010210                           MOV     R2,(R0)         ;XFR "ALU RESULT REG." TO DVMSR
000036  000036  105746                           ROMCLK                  ;CLOCK INSTRUCTION
000040  000040  005005                           CLR     R5              ;ZERO "EXPECTED" LOC
000042  000042  013704  161142                   MOV     @DVRIC,R4       ;READ "ALU RESULT"
000046  000046  001401                           BEQ     1$              ;S/B=0
000050  000050  000000                           HLT     6               ;ALU RESULT NOT=0 ON INIT
000052  000052  012710  010037           1$:     MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
000056  000056  105746                           ROMCLK                  ;DO "ALU F=A"
000060  000060  010210                           MOV     R2,(R0)         ;XFR "ALU RESULT" TO DVMSR
000062  000062  105746                           ROMCLK                  ;CLOCK INSTR
000064  000064  013704  161120                   MOV     @DVRIC,R4       ;READ RESULT
000070  000070  001401                           BEQ     2$              ;S/B=0
000072  000072  000000                           HLT     6               ;"A" REG NOT=0 ON INIT
000074  000074  012710  010026           2$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
000100  000100  105746                           ROMCLK                  ;ALU F=A+B
000102  000102  010210                           MOV     R2,(R0)         ;XFR TO RIC
000104  000104  105746                           ROMCLK                  ;CLOCK INSTR.
000106  000106  013704  161076                   MOV     @DVRIC,R4       ;READ RESULT
000112  000112  001401                           BEQ     3$              ;S/B=0
000114  000114  000000                           HLT     6               ;"B" REG NOT=0 ON INIT
000116  000116  104400           3$:     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     1     1     0     0     1    F=-1
                                        ;*   1     0     1     1     0     0     0    F=0
                                        ;*   1     0     1     1     0     1     0    F=A
                                        ;*   1     0     1     0     0     1     0    F=B
                                        ;*   1     0     1     1     0     1     1    F=A+1
                                        ;*   1     0     1     0     0     0     0    F=A+B
                                        ;;*************************************************************

                                :  TEST 75
                                ;--------------
000276  000276  012737  000075  001206  TST75:  MOV     #75,ISTNO
000304  000304  012737  021205  001216           MOV     #TST76,NEXT
                                        ;*FUNCTION TESTED
                                        ;*F=-1,RIC+F
                                        ;*
000312  000312  012737  020320  001220           MOV     #10$,LOCK       ;SET FOR SW09
000320  000320  105746                   10$:    MSTCLR                  ;CLEAR DV11
000322  000322  012777  000010  161032           MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
000330  000330  012700  001400                   MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
000334  000334  012702  030366                   MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1,R2
000340  000340  012710  010037                   MOV     #ALU+BIT4+BIT3+BIT2,(R0)
```

```
                                                RUMCLK                  ;DO "F=-1"
                                                MOV     R2,(R0)         ;XFR "ALU RESULT TO DVRIC"
                                                ROMCLK                  ;
                                                MOV     @DVRIC,R4       ;READ RESULTS
                                                MOV     #-1,R5          ;SET EXPECTED
                                                CMP     R5,R4           ;DID F=-1 WORK?
                                                BEQ     63$             ;BR IF YES
                                                HLT     6               ;F=-1 APPEARED TO FAIL
                                                ;*TEST THAT DVRIC (NOW THAT ITS ALL 1'S)
                                                ;*CAN BE CLEARED BY A MSTCLR.
                                                ;*
                                        63$:    SCOP1                   ;SW09=1?
                                                MOV     #11$,LOCK       ;SET RETURN IF SW09=1
                                        11$:    MSTCLR                  ;CLEAR DV11
                                                CLR     R5              ;SET EXPECTED RESULTS
                                                MOV     @DVRIC,R4       ;READ DVRIC
                                                BEQ     64$             ;BR IF=0
                                                HLT     6               ;DVRIC REG. NOT CLEARED ON INIT
                                                ;*NEXT SET OF FUNCTIONS:
                                                ;*F=0,RIC=F
                                                ;*
                                        64$:    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                MOV     #ALU=BIT4=BIT3=BIT2,(R0)
                                                ROMCLK                  ;"F=-1"
                                                MOV     #ALU=BIT3=BIT2,(R0)
                                                ROMCLK                  ;"F=0"
                                                MOV     R2,(R0)         ;XFR "F" TO DVRIC
                                                ROMCLK                  ;
                                                CLR     R5              ;SET EXPECTED
                                                MOV     @DVRIC,R4       ;READ RESULTS
                                                BEQ     65$             ;BR IF=0
                                                HLT     6               ;"F=0" APPEARED TO FAIL
                                                ;*NEXT SET OF FUNCTIONS:
                                                ;*F=A+1,RIC=F,A=F,F=0,F=A
                                                ;*
                                        65$:    SCOP1                   ;SW09=1?
                                                MOV     #12$,LOCK       ;SET RETURN
                                                MOV     #1,R5           ;SET GOOD RESULTS
                                        12$:    MOV     #ALU=BIT5=BIT4=BIT3=BIT2=BIT1=BIT0,(R0)
                                                BIS     #BIT1,@DVSCR    ;ISSUE ROM CLOCK
                                                MOV     R2,(R0)         ;XFR "F" TO DVRIC
                                                BIS     #BIT1,@DVSCR    ;ISSUE ROM CLK
                                                MOV     @DVRIC,R4       ;READ RESULTS
                                                CMP     R5,R4           ;FUNCTION WORK?
                                                BEQ     66$             ;YES
                                                HLT     6               ;F=A+1 APPEARED TO FAIL
                                        66$:    MOV     #XFR=BIT7=BIT5=BIT4=BIT0,(R0)
                                                BIS     #BIT1,@DVSCR    ;ISSUE ROM CLK
                                                MOV     #ALU=BIT3=BIT2,(R0)
                                                BIS     #BIT1,@DVSCR    ;ROM CLK
                                                MOV     R2,(R0)         ;XFR RIC=F
                                                BIS     #BIT1,@DVSCR    ;ROM CLK
                                                MOV     @DVRIC,R4       ;READ RESULT
                                                BEQ     67$             ;BR IF GOOD
                                                HLT     0               ;F=0 FAILED
                                        67$:    MOV     #ALU=BIT4=BIT3=BIT2=BIT1=BIT0,(R0)
```

```
                                                BIS     #BIT1,@DVSCR    ;CLOCK "F=A"
                                                MOV     R2,(R0)         ;XFR RIC←F
                                                BIS     #BIT1,@DVSCR    ;ROM CLK
                                                MOV     @DVRIC,R4       ;READ RESULT
                                                CMP     R5,R4           ;BR IF OK
                                                BEQ     69$             ;
                                                HALT                    ;F=A FAILED
                                        69$:    INC     R5              ;UPDATE DATA
                                                BNE     13$             ;BR IF NOT ALL DONE.
                                                SCOP1                   ;SW09=1?
                                                ;←NEXT SET OF FUNCTIONS:
                                                ;←F=A←1,A←F,B←F,F=A←B,RIC←F
                                                ;←
                                                MOV     #13$,LOCK       ;SET RETURN
                                        13$:    MSTCLR                  ;RESET DV11
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
                                                MOV     #0,R5           ;SET EXPECTED RESULTS
                                                MOV     @DVRA1,R1       ;SET POINTER
                                                MOV     @DVSCR,R3       ;SET POINTER
                                                MOV     #ALU←BIT5←BIT4←BIT3←BIT2←BIT1←BIT0,(R0)
                                                ROMCLK                  ;F=A←1
                                                MOV     #XFR←BIT7←BIT6←BIT5←BIT4←BIT0,(R0)
                                                ROMCLK                  ;XFR A←ALU RESULT
                                                MOV     #XFR←BIT7←BIT6←BIT5←BIT4←BIT1,(R0)
                                        23:     BIS     #BIT1,(R3)      ;XFR B←ALU RESULT
                                                MOV     #ALU←BIT4←BIT2←BIT1,(R0)
                                                BIS     #BIT1,(R3)      ;CLOCK F=A←B
                                                MOV     R0,(R0)         ;XFR RIC←F
                                                BIS     #BIT1,(R3)      ;
                                                MOV     @DVRIC,R4       ;READ DVRIC
                                                CMP     R5,R4           ;FUNCTION WORK?
                                                BEQ     63$             ;BR IF YES
                                                HALT                    ;F=A←B APPEARED TO FAIL
                                        63$:    INC     R0              ;INC DATA POINTER
                                                CMP     #0,R5           ;ALL DONE?
                                                BNE     23$             ;BR IF NO
                                                MSTCLR                  ;RESET DV11
                                                MOV     @DVRIC,R4       ;DVRIC ZERO ON INIT
                                                BEQ     70$             ;BR IF YES
                                                HALT                    ;R0=0
                                        70$:    STOP1                   ;SW09=1?
                                                ;←NEXT SET OF FUNCTIONS:
                                                ;←F=-1,B←F,F=0,F=B
                                                ;←
                                                MOV     #14$,LOCK       ;SET RETURN
                                        14$:    MSTCLR                  ;RESET DV11
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
                                                MOV     #ALU←BIT4←BIT3←BIT2,(R0)
                                                ROMCLK                  ;F=-1
                                                MOV     #XFR←BIT7←BIT6←BIT5←BIT4←BIT1,(R0)
                                                ROMCLK                  ;XFR B←F
                                                MOV     #ALU←BIT3←BIT2,(R0) ;
                                                ROMCLK                  ;F=0
                                                MOV     #ALU←BIT2←BIT0,(R0) ;
                                                ROMCLK                  ;F=0
                                                MOV     R2,(R0)         ;XFR RIC←F
```

```
021030  104415                          ROMCLK                  ;CLOCK
021032  017104  177776                  MOV     @DVRIC,R4       ;READ RESULTS
021036  012705  177777                  MOV     #-1,R5          ;SET EXPECTED
021042  020504                          CMP     R5,R4           ;DID F=B WORK?
021044  001407                          BEQ     71$             ;BR IF YES
021046  104000                          HLT     0               ;F=B FAILED
021050  104401                  71$:    SCOPE                   ;SWD9=1?
                                        ;*NEXT SET OF FUNCTIONS;
                                        ;*CATCH "SET/CLEAR" THAT WAS MISSED.
                                        ;*F=-1.S/CIALUDI=D1,RIC=F
                                        ;*
021054  012737  021062  001220          MOV     #15$,LOCK       ;SET RETURN
021062  104410                  15$:    MSTCLR                  ;RESET DV11
021064  012737  000010  160270          MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
021072  010034                          MOV     #ALU+BIT4+BIT3+BIT2,(R0)
021074  104410                          ROMCLK                  ;F=-1
021076  012710  050026                  MOV     #S.C+BIT4+BIT2+BIT1,(R0)
021102  104410                          ROMCLK                  ;S/C "CLEAR ALUDI"
021104  010210                          MOV     R2,(R0)         ;XFR RIC=F
021106  104410                          ROMCLK                  ;
021110  011104                          MOV     (R1),R4         ;READ DVRIC
021112  012705  177775                  MOV     #177775,R5      ;SET EXPECTED
021116  020504                          CMP     R5,R4           ;DID S/C ALUDI WORK?
021120  001407                          BEQ     72$             ;BR IF YES
021122  104000                          HLT     0               ;S/C ALUDI FAILED.
021124  104401                  72$:    SCOPE                   ;SWD9=1?
                                        ;*NEXT SET OF FUNCTIONS;
                                        ;*CATCH ANOTHER "SET/CLEAR" THAT WAS MISSED.
                                        ;*F=-1.S/CIALU HIGH BYTE=D1,RIC=F
                                        ;*
021130  012737  021136  001220          MOV     #16$,LOCK       ;SET RETURN
021136  104410                  16$:    MSTCLR                  ;RESET DV11
021140  012737  000010  160214          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
021146  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
021152  104410                          ROMCLK                  ;F=-1
021154  012710  050027                  MOV     #S.C+BIT4+BIT3+BIT1+BIT0,(R0)
021160  104410                          ROMCLK                  ;S/C "CLEAR ALU HIGH BYTE"
021162  010210                          MOV     R2,(R0)         ;XFR RIC=F
021164  104410                          ROMCLK                  ;
021166  011104                          MOV     (R1),R4         ;READ RIC
021170  012705  000377                  MOV     #377,R5         ;SET EXPECTED
021174  020504                          CMP     R5,R4           ;DID S/C WORK?
021176  001407                          BEQ     73$             ;BR IF YES
021200  104000                          HLT     0               ;S/C ALU HIGH BYTE FAILED.
021202  104401                  73$:    SCOPE;                  ;SWD9=1?
021204  104400                          SCOPE                   ;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
                                        ;--------------
021206  012737  000076  001606  ST76:   MOV     #76,ISTNO
021214  012737  021334  001216          MOV     #TST77,NEXT
021222  104420                          MSTCLR                  ;RESET DV11
021226  012737  000010  160130          MOV     #BIT3,DVSCR     ;SET SOURCE SEL
021234  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER
021240  012710  032126                  MOV     #XFR+BIT6+BIT4+BIT2+BIT1,(R0
021244  010210                          MOV     R2,(R0)         ;XFR "MASTER SCAN 0-3" TO DVRIC
021246  104410                          ROMCLK                  ;
021250  005005                          CLR     R5              ;SET EXPECTED
021252  013704  160112                  MOV     DVRIC,R4        ;READ RESULTS
021256  001401                          BEQ     1$              ;BR IF=0
021260  005204                          INC     R4              ;MSCAN NOT=0 ON INIT
021262  005205                  1$:     INC     R5              ;UPDATE POINTER
021264  012703  000035                  MOV     #35,R3          ;SET COUNT TO 35
021270  010502                  2$:     MOV     #S.C+BIT6+BIT1,(R0) ;S/C "ADVANCE MSCAN"
021272  104410                          ROMCLK                  ;CLOCK
021274  010502                          MOV     #S.C+BIT6+BIT1,(R0) ;S/C "ADVANCE MSCAN"
021276  104410                          ROMCLK                  ;CLOCK
021300  010210                          MOV     R2,(R0)         ;XFR RIC=MSCAN
021302  104410                          ROMCLK                  ;CLOCK
021304  013704  160054                  MOV     DVRIC,R4        ;READ RESULTS
021310  020405                          CMP     R5,R4           ;MSCAN INCREMENTED?
021312  001401                          BEQ     3$              ;BR IF YES
021314  104000                          HALT                    ;MSCAN WRONG
021316  005205                  3$:     INC     R5              ;UPDATE
021320  005303                          DEC     R3              ;CLEAN
021322  177760                          BNE     2$              ;COUNT DONE?
021324  005703                          TST     R3              ;BR IF NO
021326  001355                          BNE     2$              ;SCOPE
021330  104402                          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 DVRIC REG.
                                        ;;**********************************************************

                                        ;  TEST 77
                                        ;--------------
021334  012737  000077  001606  ST77:   MOV     #77,TSTNO
021342  012737  021460  001216          MOV     #TST100,NEXT
021350  104410                          MSTCLR                  ;RESET DV11
021352  005000                          CLR     R0              ;
021354  005001                          CLR     R1              ;
021356  013702  001372                  MOV     DVSRS,R2        ;
021362  013703  001374                  MOV     DVSRSH,R3       ;
021366  013705  001370                  MOV     DVSRA,R5        ;
021372  012704  000001                  MOV     #1,R4           ;
021376  110013                  1$:     MOVB    R0,(R3)         ;LOAD LINE NUMBER
021400  110113                          MOVB    R1,(R3)         ;LOAD SEC. REG. POINTER
021402  010415                          MOV     R4,(R5)         ;LOAD DATA
021404  122024                          CMPB    (R0)+,(R4)+     ;UPDATE LINE AND DATA
021406  020027  000020                  CMP     #16.,R0         ;ALL LINES DONE?
```

```
021412  001371                  BNE     1$                      ;BR IF NO
021414  005000                  CLR     R0                      ;ZERO LINE POINTER
021416  005201                  INC     R1                      ;UPDATE SEC REG POINTER.
021420  022701  000020          CMP     #16.,R1                 ;ALL SEC REG POINTERS DONE?
021424  001354                  BNE     1$                      ;BR IF NO
021426  005067  157740      2$: CLR     @DVSRS                  ;ZERO POINTERS
021432  012705  000001          MOV     #1,R5                   ;SET GOOD DATA
021436  012703  000010  157716  MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
021442  012703  000000          MOV     #RAM,R3                 ;LOAD RAM INSTR.
021446  016700  000400          MOV     DVSFR,R0                ;SET POINTER
021452  012700  030266          MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT1,R2
021456  010310              3$: MOV     R3,(R0)                 ;DO RAM READ
021460  104400                  ROMCLK                          ;EXECUTE
021462  010210                  MOV     R2,(R0)                 ;XFR RIC+RAM OUTPUT
021464  104400                  ROMCLK                          ;CLOCK
021466  016704  157672          MOV     @DVRIC,R4               ;READ RESULT
021472  020405                  CMP     R5,R4                   ;GOOD?
021474  001401                  BEQ     4$                      ;BR IF YES
021476  000000                  HALT                            ;RAM READ FAILED.
021500  062705  000020      4$: ADD     #20,R5                  ;UPDATE DATA
021504  005203                  INC     R3                      ;UPDATE POINTER
021506  022703  000080          CMPB    #16.,R3                 ;ALL DONE
021512  001361                  BNE     3$                      ;BR IF NO
021514  042705  177760          BIC     #^C<17>,R5              ;CLEAR JUNK
021520  012703  020000          MOV     #RAM,R3                 ;SET RAM INSTR
021524  010046                  MOV     R0,-(SP)                ;SAVE R0 ON STACK
021526  004537  031342          PERFORM .SETSCAN                ;UPDATE MSCANNER
                                ;
021532  012600                  MOV     (SP)+,R0                ;RESTORE R0
021534  005205                  INC     R5                      ;UPDATE DATA
021536  012710  030124          MOV     #XFR+BIT6+BIT4+BIT2,(R0) ;XFR RAR-MSCAN 0-3
021542  104400                  ROMCLK                          ;EXECUTE
021544  022705  000020          CMP     #16.,R5                 ;ALL DONE?
021550  001341                  BNE     3$                      ;BR IF NO
021552  104400                  SCOPE                           ;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 16=1.0
                                ;*  1     0       0       0     ALU 13-  =1.0
                                ;*                              -12=1.0
                                ;*  1     0       1       0     ALU 00=1.0
                                ;*  1     0       1       1     ALU 01=1.0
                                ;*  1     1       0       0     ALU 02=1.0
                                ;*  1     1       0       1     ALU 03=1.0
                                ;*  1     1       1       0     ALU 04=1.0
                                ;;*********************************************************

                        ;  TEST 100
                        ;--------------
021554  016737  000100  001826  TST100: MOV     #100,TSTNO
021562  012737  023162  001216          MOV     #TST101,NEXT
```

```
4417  021574  005077  157572                    CLR     @DVSR5
4418                                            :*BRANCH "A" TEST OF ALU 15
4419                                            :*
4420  021600  104412                            MSTCLR                  ;RESET DV11
4421  021602  012777  000010  157558            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4422  021610  012777  100000  157560            MOV     #BIT15,@DVSRA   ;LOAD DATA
4423  021616  012777  020000  157554            MOV     #RAM,@DVSFR     ;DO A "RAM READ"
4424  021624  104415                            ROMCLK                  ;EXECUTE
4425  021626  012777  030261  157544            MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4426  021634  104415                            ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
4427  021636  012777  010037  157534            MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4428  021644  104415                            ROMCLK                  ;F=A
4429  021646  012777  000000  157524            MOV     #0000,@DVSFR    ;LOAD BRANCH POINT
4430  021654  017704  157510                    MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
4431  021660  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4432  021664  012705  000002                    MOV     #BIT1,R5        ;SET EXPECTED
4433  021670  020504                            CMP     R5,R4           ;BR POINTS CORRECT?
4434  021672  001401                            BEQ     64$             ;BR IF YES
4435  021674  104006                            HLT     6               ;BRANCH POINTS WRONG
4436  021676  104412                    64$:    MSTCLR
4437  021700  012777  000010  157454            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
4438  021706  012777  000000  157464            MOV     #0000,@DVSFR    ;RESET DV11
4439  021714  017704  157450                    MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
4440  021720  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4441  021724  012705  000003                    MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
4442  021730  020504                            CMP     R5,R4           ;SET EXPECTED
4443  021732  001401                            BEQ     65$             ;BR POINT OK?
4444  021734  104006                            HLT     6               ;BR IF YES
4445  021736                            65$:    ;BR POINTS WRONG
4446                                            :*BRANCH "A" TEST OF ALU 13-
4447                                            :*
4448  021736  104412                            MSTCLR                  ;RESET DV11
4449  021740  012777  000010  157414            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4450  021746  012777  020000  157422            MOV     #BIT13,@DVSRA   ;LOAD DATA
4451  021754  012777  020000  157416            MOV     #RAM,@DVSFR     ;DO A "RAM READ"
4452  021762  104415                            ROMCLK                  ;EXECUTE
4453  021764  012777  030261  157406            MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4454  021772  104415                            ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
4455  021774  012777  010037  157376            MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4456  022002  104415                            ROMCLK                  ;F=A
4457  022004  012777  004000  157366            MOV     #BIT11,@DVSFR   ;LOAD BRANCH POINT
4458  022012  017704  157352                    MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
4459  022016  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4460  022022  012705  000002                    MOV     #BIT1,R5        ;SET EXPECTED
4461  022026  020504                            CMP     R5,R4           ;BR POINTS CORRECT?
4462  022030  001401                            BEQ     66$             ;BR IF YES
4463  022032  104006                            HLT     6               ;BRANCH POINTS WRONG
4464  022034  104412                    66$:    MSTCLR
4465  022036  012777  000010  157316            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
4466  022044  012777  004000  157326            MOV     #BIT11,@DVSFR   ;RESET DV11
4467  022052  017704  157312                    MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
4468  022056  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4469  022062  012705  000003                    MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
4470  022066  020504                            CMP     R5,R4           ;SET EXPECTED
4471  022070  001401                            BEQ     67$             ;BR POINT OK?
4472  022072  104006                            HLT     6               ;BR IF YES
```

```
4473  022074                          67$:                    ;BR POINTS WRONG
4474                                      :*BRANCH "A" TEST OF ALU   -12
4475                                      :*
4476  022074  104412                    MSTCLR                  ;RESET DV11
4477  022076  012777  000010  157256    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4478  022104  012777  010000  157264    MOV     #BIT12,@DVSRA   ;LOAD DATA
4479  022112  012777  020000  157260    MOV     #RAM,@DVSFR     ;DO A "RAM READ"
4480  022120  104415                    ROMCLK                  ;EXECUTE
4481  022122  012777  030261  157250    MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4482  022130  104415                    ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
4483  022132  012777  010037  157240    MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4484  022140  104415                    ROMCLK                  ;F=A
4485  022142  012777  004000  157230    MOV     #BIT11,@DVSFR   ;LOAD BRANCH POINT
4486  022150  017704  157214            MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
4487  022154  042704  177774            BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4488  022160  012705  000002            MOV     #BIT1,R5        ;SET EXPECTED
4489  022164  020504                    CMP     R5,R4           ;BR POINTS CORRECT?
4490  022166  001401                    BEQ     69$             ;BR IF YES
4491  022170  104006                    HLT     6               ;BRANCH POINTS WRONG
4492  022172  104412            69$:    MSTCLR
4493  022174  012777  000010  157160    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
4494  022202  012777  004000  157170    MOV     #BIT11,@DVSFR   ;RESET DV11
4495  022210  017704  157154            MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
4496  022214  042704  177774            BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4497  022220  012705  000003            MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
4498  022224  020504                    CMP     R5,R4           ;SET EXPECTED
4499  022226  001401                    BEQ     69$             ;BR POINT OK?
4500  022230  104006                    HLT     6               ;BR IF YES
4501  022232                          69$:                    ;BR POINTS WRONG
4502                                      :*BRANCH "A" TEST OF ALU 00
4503                                      :*
4504  022232  104412                    MSTCLR                  ;RESET DV11
4505  022234  012777  000010  157120    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4506  022242  012777  000001  157126    MOV     #BIT0,@DVSRA    ;LOAD DATA
4507  022250  012777  020000  157122    MOV     #RAM,@DVSFR     ;DO A "RAM READ"
4508  022256  104415                    ROMCLK                  ;EXECUTE
4509  022260  012777  030261  157112    MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4510  022266  104415                    ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
4511  022270  012777  010037  157102    MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4512  022276  104415                    ROMCLK                  ;F=A
4513  022300  012777  005000  157072    MOV     #BIT11+BIT9,@DVSFR      ;LOAD BRANCH POINT
4514  022306  017704  157056            MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
4515  022312  042704  177774            BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4516  022316  012705  000002            MOV     #BIT1,R5        ;SET EXPECTED
4517  022322  020504                    CMP     R5,R4           ;BR POINTS CORRECT?
4518  022324  001401                    BEQ     70$             ;BR IF YES
4519  022326  104006                    HLT     6               ;BRANCH POINTS WRONG
4520  022330  104412            70$:    MSTCLR
4521  022332  012777  000010  157022    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
4522  022340  012777  005000  157032    MOV     #BIT11+BIT9,@DVSFR      ;RESET DV11
4523  022346  017704  157016            MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
4524  022352  042704  177774            BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4525  022356  012705  000003            MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
4526  022362  020504                    CMP     R5,R4           ;SET EXPECTED
4527  022364  001401                    BEQ     71$             ;BR POINT OK?
4528  022366  104006                    HLT     6               ;BR IF YES
```

```
 4529  022370                          71$:                    ;BR POINTS WRONG
 4530                                                          ;*BRANCH "A" TEST OF ALU 01
 4531                                                          ;*
 4532  022370  104412                                        MSTCLR                  ;RESET DV11
 4533  022372  012777  000010  156762                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 4534  022400  012777  000002  156770                        MOV     #BIT1,@DVSRA    ;LOAD DATA
 4535  022406  012777  020000  156764                        MOV     #RAM,@DVSFR     ;DO A "RAM READ"
 4536  022414  104415                                        ROMCLK                  ;EXECUTE
 4537  022416  012777  030261  156754                        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
 4538  022424  104415                                        ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
 4539  022426  012777  010037  156744                        MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
 4540  022434  104415                                        ROMCLK                  ;F=A
 4541  022436  012777  005400  156734                        MOV     #BIT11+BIT9+BIT8,@DVSFR ;LOAD BRANCH POINT
 4542  022444  017704  156720                              MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
 4543  022450  042704  177774                              BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4544  022454  012705  000002                              MOV     #BIT1,R5        ;SET EXPECTED
 4545  022460  020504                                      CMP     R5,R4           ;BR POINTS CORRECT?
 4546  022462  001401                                      BEQ     72$             ;BR IF YES
 4547  022464  104006                                      HLT     6               ;BRANCH POINTS WRONG
 4548  022466  104412                          72$:        MSTCLR
 4549  022470  012777  000010  156664                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 4550  022476  012777  005400  156674                        MOV     #BIT11+BIT9+BIT8,@DVSFR ;RESET DV11
 4551  022504  017704  156660                              MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
 4552  022510  042704  177774                              BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
 4553  022514  012705  000003                              MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
 4554  022520  020504                                      CMP     R5,R4           ;SET EXPECTED
 4555  022522  001401                                      BEQ     73$             ;BR POINT OK?
 4556  022524  104006                                      HLT     6               ;BR IF YES
 4557  022526                          73$:                    ;BR POINTS WRONG
 4558                                                          ;*BRANCH "A" TEST OF ALU 02
 4559                                                          ;*
 4560  022526  104412                                        MSTCLR                  ;RESET DV11
 4561  022530  012777  000010  156624                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 4562  022536  012777  000004  156632                        MOV     #BIT2,@DVSRA    ;LOAD DATA
 4563  022544  012777  020000  156626                        MOV     #RAM,@DVSFR     ;DO A "RAM READ"
 4564  022552  104415                                        ROMCLK                  ;EXECUTE
 4565  022554  012777  030261  156616                        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
 4566  022562  104415                                        ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
 4567  022564  012777  010037  156606                        MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
 4568  022572  104415                                        ROMCLK                  ;F=A
 4569  022574  012777  006000  156576                        MOV     #BIT11+BIT10,@DVSFR     ;LOAD BRANCH POINT
 4570  022602  017704  156562                              MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
 4571  022606  042704  177774                              BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4572  022612  012705  000002                              MOV     #BIT1,R5        ;SET EXPECTED
 4573  022616  020504                                      CMP     R5,R4           ;BR POINTS CORRECT?
 4574  022620  001401                                      BEQ     74$             ;BR IF YES
 4575  022622  104006                                      HLT     6               ;BRANCH POINTS WRONG
 4576  022624  104412                          74$:        MSTCLR
 4577  022626  012777  000010  156526                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 4578  022634  012777  006000  156536                        MOV     #BIT11+BIT10,@DVSFR     ;RESET DV11
 4579  022642  017704  156522                              MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
 4580  022646  042704  177774                              BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
 4581  022652  012705  000003                              MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
 4582  022656  020504                                      CMP     R5,R4           ;SET EXPECTED
 4583  022660  001401                                      BEQ     75$             ;BR POINT OK?
 4584  022662  104006                                      HLT     6               ;BR IF YES
```

```
 4585  022664                          75$:                    ;BR POINTS WRONG
 4586                                      :*BRANCH "A" TEST OF ALU 03
 4587                                      :*
 4588  022664  104412                      MSTCLR                  ;RESET DV11
 4589  022666  012777  000010  156466      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 4590  022674  012777  000010  156474      MOV     #BIT3,@DVSRA    ;LOAD DATA
 4591  022702  012777  020000  156470      MOV     #RAM,@DVSFR     ;DO A "RAM READ"
 4592  022710  104415                      ROMCLK                  ;EXECUTE
 4593  022712  012777  030261  156460      MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
 4594  022720  104415                      ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
 4595  022722  012777  010037  156450      MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
 4596  022730  104415                      ROMCLK                  ;F=A
 4597  022732  012777  006400  156440      MOV     #BIT11+BIT10+BIT8,@DVSFR        ;LOAD BRANCH POINT
 4598  022740  017704  156424              MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
 4599  022744  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4600  022750  012705  000002              MOV     #BIT1,R5        ;SET EXPECTED
 4601  022754  020504                      CMP     R5,R4           ;BR POINTS CORRECT?
 4602  022756  001401                      BEQ     76$             ;BR IF YES
 4603  022760  104006                      HLT     6               ;BRANCH POINTS WRONG
 4604  022762  104412          76$:        MSTCLR
 4605  022764  012777  000010  156370      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 4606  022772  012777  006400  156400      MOV     #BIT11+BIT10+BIT8,@DVSFR        ;RESET DV11
 4607  023000  017704  156364              MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
 4608  023004  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
 4609  023010  012705  000003              MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
 4610  023014  020504                      CMP     R5,R4           ;SET EXPECTED
 4611  023016  001401                      BEQ     77$             ;BR POINT OK?
 4612  023020  104006                      HLT     6               ;BR IF YES
 4613  023022                          77$:                    ;BR POINTS WRONG
 4614                                      :*BRANCH "A" TEST OF ALU 04
 4615                                      :*
 4616  023022  104412                      MSTCLR                  ;RESET DV11
 4617  023024  012777  000010  156330      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 4618  023032  012777  000020  156336      MOV     #BIT4,@DVSRA    ;LOAD DATA
 4619  023040  012777  020000  156332      MOV     #RAM,@DVSFR     ;DO A "RAM READ"
 4620  023046  104415                      ROMCLK                  ;EXECUTE
 4621  023050  012777  030261  156322      MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
 4622  023056  104415                      ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
 4623  023060  012777  010037  156312      MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
 4624  023066  104415                      ROMCLK                  ;F=A
 4625  023070  012777  007000  156302      MOV     #BIT11+BIT10+BIT9,@DVSFR        ;LOAD BRANCH POINT
 4626  023076  017704  156266              MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
 4627  023102  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4628  023106  012705  000002              MOV     #BIT1,R5        ;SET EXPECTED
 4629  023112  020504                      CMP     R5,R4           ;BR POINTS CORRECT?
 4630  023114  001401                      BEQ     78$             ;BR IF YES
 4631  023116  104006                      HLT     6               ;BRANCH POINTS WRONG
 4632  023120  104412          78$:        MSTCLR
 4633  023122  012777  000010  156232      MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 4634  023130  012777  007000  156242      MOV     #BIT11+BIT10+BIT9,@DVSFR        ;RESET DV11
 4635  023136  017704  156226              MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
 4636  023142  042704  177774              BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
 4637  023146  012705  000003              MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
 4638  023152  020504                      CMP     R5,R4           ;SET EXPECTED
 4639  023154  001401                      BEQ     79$             ;BR POINT OK?
 4640  023156  104006                      HLT     6               ;BR IF YES
```

```
 4641  023160                          79$:                            ;BR POINTS WRONG
 4642  023160  104400                        SCOPE
 4643
 4644
 4645                                        ;************************ TEST 101 *****************************
 4646                                        ;*TEST OF BRANCH "B" "RAM OUTPUT 0-14=0".
 4647                                        ;*TEST TO A RAM READ AND "FLOAT" A "1" FROM
 4648                                        ;*RAM 0 TO 14 ; EXPECTING "RAM 014=0" TO BE FALSE.
 4649                                        ;*THEN THE "1" IS SHIFTED INTO BIT15 AND
 4650                                        ;*"RAM 0-14=0" SHOULD BE FALSE.
 4651                                        ;*THIS ALSO TEST "BRB" [RAM OUTPUT BIT15] TRUE.
 4652                                        ;;*************************************************************
 4653
 4654                                ;  TEST 101
 4655                                ;--------------
 4656  023162  012737  000101  001226 TST101: MOV     #101,TSTNO
 4657  023170  012737  023362  001216         MOV     #TST102,NEXT
 4658  023176  104412                         MSTCLR                  ;RESET DV11
 4659  023200  012703  000001                 MOV     #1,R3           ;SET DATA
 4660  023204  012702  076000                 MOV     #BRB+BIT11+BIT10,R2 ;BRB "RAM OUTPUT 0-14=0"?
 4661  023210  012777  000010  156144         MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 4662  023216  010277  156156                 MOV     R2,@DVSFR       ;LOAD BRB TEST
 4663  023222  017704  156142                 MOV     @DVLCR,R4       ;READ TEST POINTS
 4664  023226  012705  000001                 MOV     #BIT0,R5        ;SET EXPECTED
 4665  023232  042704  177774                 BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4666  023236  020504                         CMP     R5,R4           ;TRUE?
 4667  023240  001401                         BEQ     .+4             ;BR IF YES
 4668  023242  104006                         HLT     6               ;RAM OUTPUT 0-14 NOT TRUE AFTER INIT
 4669  023244  012705  000003                 MOV     #BIT1+BIT0,R5   ;SET EXPECTED
 4670  023250  010377  156122          1$:    MOV     R3,@DVSRA       ;LOAD DATA
 4671  023254  012777  020000  156116         MOV     #RAM,@DVSFR     ;ISSUE RAM READ INSTR
 4672  023262  104415                         ROMCLK                  ;
 4673  023264  010277  156110                 MOV     R2,@DVSFR       ;BRB TEST
 4674  023270  017704  156074                 MOV     @DVLCR,R4       ;READ BR TEST POINTS
 4675  023274  042704  177774                 BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4676  023300  005703                         TST     R3              ;IS 15=1 IN DATA?
 4677  023302  100002                         BPL     2$              ;BR IF NO
 4678  023304  042705  000002                 BIC     #BIT1,R5        ;IF 15=1 - 0-14=TRUE
 4679  023310  020504                  2$:    CMP     R5,R4           ;BRB TEST POINT OK?
 4680  023312  001401                         BEQ     3$              ;BR IF YES
 4681  023314  104006                         HLT     6               ;BAD TEST POINT
 4682  023316  000241                  3$:    CLC                     ;CLEAR CARRY
 4683  023320  006103                         ROL     R3              ;MOV BIT TO NEXT RAM POSITION
 4684  023322  001352                         BNE     1$              ;HAVE ALL BITS BEEN TESTED?
 4685  023324  012705  000001                 MOV     #BIT0,R5        ;SET EXPECTED RESULTS
 4686  023330  012777  075400  156042         MOV     #BRB+BIT11+BIT9+BIT8,@DVSFR
 4687  023336  017704  156026                 MOV     @DVLCR,R4       ;BRB "RAM OUTPUT 15H"
 4688  023342  042704  177774                 BIC     #^C<BIT1+BIT0>,R4
 4689  023346  020504                         CMP     R5,R4           ;BRANCH RESULTS OK?
 4690  023350  001403                         BEQ     4$              ;BR IF YES
 4691  023352  017702  156022                 MOV     @DVSFR,R2       ;SAVE DVSFR FOR TYPEOUT
 4692  023356  104006                         HLT     6               ;"RAM OUTPUT 15H" S/B TRUE
 4693  023360  104400                  4$:    SCOPE                   ;SCOPE TESTS
 4694
 4695
 4696                                        ;************************ 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)        ;SAVERS
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
```

```
                                                        MOV     R2,-(SP)        ;FALSE!
                                                        MOV     #DVSFR,R2
                                                        MOV     #BIT1+BIT0,R5   ;EXPECTED
                                                        MOV     #DVLCR,R4       ;READ RESULTS
                                                        BIC     #^C(BIT1+BIT0),R4 ;STRIP JUNK
                                                        CMP     R5,R4           ;GOOD?
                                                        BEQ     .+4             ;
                                                        INC     (SP)            ;WRITE INHIBIT S/B FALSE
                                                        MOV     (SP)+,R2        ;RESTORE
                                                        MOV     (SP)+,R4        ;REGISTERS
                                                        MOV     (SP)+,R5        ;
                                                        MOV     R4,(PC)         ;MOVE ALU RESULT TO A REG (UPDATED DATA)
                                                        ROMCLK                  ;EXECUTE
                                                        INC     R1              ;UPDATE SEC REG POINTER
                                                        CMP     #16.,R1         ;ALL REGISTERS DONE?
                                                        BNE     .+4             ;BR IF NO
                                                        MOV     R0,-(SP)        ;SAVE R0
                                                        PERFORM .SETSCAN        ;UPDATE MSCAN
                                                        .                       ;(LINE NO. UPDATE)
                                                        MOV     (SP)+,R0        ;RESTORE R0
                                                        MOV     #XFR+BIT6+BIT4+BIT2,(R0)
                                                        ROMCLK                  ;XFR MSCAN TO RAM C-3 CLOCK
                                                        INC     R5              ;UPDATE LINE NUMBER COUNTER.
                                                        CMP     #16.,R5         ;ALL LINES DONE?
                                                        BNE     .+4             ;BR IF NO
                                                        CLR     R0              ;CLEAR SEC REG POINTER
                                                        CLR     @TEMP           ;CLEAR LINE NUMBER POINTER
                                                        MOV     DVSRS,R2        ;SET SRC POINTER (LINE SEL)
                                                        MOV     DVSRSH,R3       ;SET HIGH BYTE POINTER (SEC REG SEL)
                                                        MOV     DVSRH,R1        ;SET SRH POINTER (ACCESS REG)
                                                        MOV     #1,R5           ;SET EXPECTED DATA
                                                4$:     MOVB    R0,(R3)         ;LOAD SEC REG SEL
                                                        MOVB    @TEMP,(R2)      ;LOAD LINE NO.
                                                        MOV     (R1),R4         ;READ RAM RESULT
                                                        CMP     R5,R4           ;WAS RAM "WRITTEN" CORRECTLY?
                                                        BEQ     5$              ;BR IF RAM DATA OK
                                                        INC     (SP)            ;RAM WAS "WRITTEN" INCORRECTLY
                                                5$:     CMPB    (R0)+,(R5)+     ;UPDATE SEC REG POINTER AND DATA EXPECTED
                                                        CMP     #16.,R0         ;ALL SEC REGISTERS DONE?
                                                        BNE     4$              ;BR IF NO
                                                        CLR     R0              ;ZERO SEC REG POINTER
                                                        INC     @TEMP           ;UPDATE LINE NUMBER POINTER
                                                        CMP     @TEMP,#16.      ;ALL LINES DONE?
                                                        BNE     4$              ;BR IF NO
                                                        SCOPE                   ;SCOPE TEST

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

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

```
                                        ;*TEST OF LRC B.
                                        ;*PATTERNS ARE:
                                        ;*A REG   B REG   BCC (EXPECTED)
                                        ;*  0.S     0.S     0.
                                        ;*  0.S     1.S     1.
                                        ;*  1.S     0.S     1.
                                        ;*  1.S     1.S     0.S
                                        ;;*****************************************************************

                                ;  TEST 103
                                ;--------------
024176  016737  000103  001336  TST103: MOV     #103,ISTNO
024202  016737  024262  001216          MOV     #TST104,NEXT
024210  104412                          MSTCLR                  ;RESET DV11
024212  016777  000010  155340          MOV     #BIT3,DDVSCR    ;SET SOURCE SEL
024220  016700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER
024224  012760  030345                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
024232  012710  060000                  MOV     #BCC,(R0)       ;DO A "BCC" INSTRUCTION USING "LRC8"
024236  104411                          ROMCLK                  ;EXECUTE
024240  010210                          MOV     R2,(R0)         ;XFR BCC REG TO DVRIC
024242  104411                          ROMCLK                  ;EXECUTE
024244  005005                          CLR     R5              ;SET EXPECTED TO 0
024246  016704  155314                  MOV     DDVRIC,R4       ;READ RESULTS OF BCC REG.
024252  001401                          BEQ     1$              ;BR IF =0
024254  104401                          HLT     6               ;BCC REG NOT =0
024256  012710  010034          1$:     MOV     #ALU+BIT4+BIT3+BIT2,(R0)
024262  104411                          ROMCLK                  ;ALU OPR "F=-1"
024264  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
024270  104411                          ROMCLK                  ;XFR ALU RESULT TO B REGISTER
024272  012710  060000                  MOV     #BCC,(R0)       ;DO A "BCC" OPR
024276  104411                          ROMCLK                  ;EXECUTE
024300  010210                          MOV     R2,(R0)         ;XFR BCC REG TO DVRIC
024302  104411                          ROMCLK                  ;EXECUTE
024304  012705  000377                  MOV     #377,R5         ;SET EXPECTED RESULTS =ALL 1'S (LOW BYTE)
024310  016704  155252                  MOV     DDVRIC,R4       ;READ RESULTS
024314  020504                          CMP     R5,R4           ;DID BCC OPR WORK
024316  001401                          BEQ     2$              ;BR IF OK
024320  104406                          HLT     6               ;EITHER "BCC" OPR FAILED OR BIT(S) DROPPED IN LOW BYTE O
024322  104412                  2$:     MSTCLR                  ;RESET INTERAL REG (BCC,ALU,B)
024324  016777  000010  155230          MOV     #BIT3,DDVSCR    ;SET SOURCE SEL.
024332  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
024336  104411                          ROMCLK                  ;ALU "F=-1"
024340  012710  030361                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
024344  104411                          ROMCLK                  ;XFR ALU RESULT TO "A" REGISTER
024346  012710  060000                  MOV     #BCC,(R0)       ;BCC OPERATION
024352  104411                          ROMCLK                  ;EXECUTE
024354  010210                          MOV     R2,(R0)         ;XFR BCC TO DVRIC
024356  104411                          ROMCLK                  ;EXECUTE
024360  016704  155202                  MOV     DDVRIC,R4       ;READ RESULTS
024364  020504                          CMP     R5,R4           ;BCC REG S/B =377
024366  001401                          BEQ     3$              ;BR IF OK
024370  104406                          HLT     6               ;BCC REG NOT =377
                                                                ;READ OF DATA FROM "A" REG LEG FAILED
024372  104412                  3$:     MSTCLR                  ;CLEAR INTERNAL REGISTERS
024374  016777  000010  155160          MOV     #BIT3,DDVSCR    ;SET SOURCE SEL
024402  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
```

```
                                        RUMCLK                  ;ALU "F=-1"
                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
                                        ROMCLK                  ;XFR ALU RESULT TO "A" REG
                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
                                        ROMCLK                  ;XFR ALU RESULT TO "B" REG
                                        MOV     #BCC,(R0)       ;SO BCC OPR
                                        ROMCLK                  ;
                                        MOV     R2,(R0)         ;XFR BCC TO DVRIC
                                        ROMCLK                  ;ROMCLK
                                        CLR     R5              ;EXPECT 0'S
                                        MOV     @DVRIC,R4       ;READ BCC RESULT
                                        BEQ     4$              ;BR IF =0
                                        HLT     6               ;BCC "LRC" XOR TEST FAILED
                                4$:     SCOPE                   ;SCOPE TEST

                                        ;************************ TEST 104 ****************************
                                        ;*TEST OF POLYNOMIAL "CRC 16"
                                        ;*TEST THAT BITS 9-13 OF THE "B" REG APPEAR
                                        ;*IN BITS 1-5 OF THE BCC REG.
                                        ;;*************************************************************

                                ;  TEST 104
                                ;---------------
                                TST104: MOV     #104,ISTNO
                                        MOV     #TST105,NEXT
                                        MSTCLR                  ;RESET DV11
                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                        MOV     DVSFR,R0        ;SET DVSFR POINTER
                                        MOV     #^B1001,@DVSFR  ;LOAD "DATA"
                                        MOV     #RAM,(R0)       ;RAM READ
                                        ROMCLK                  ;EXECUTE
                                        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
                                        ROMCLK                  ;XFR RAM OUTPUT TO "A" REG.
                                        MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
                                        ROMCLK                  ;XFR RAM OUTPUT TO "B" REG.
                                        MOV     #BIT3,@DVSRA    ;SELECT POLYNOMIAL "CRC16"
                                        MOV     #RAM,(R0)       ;READ DATA
                                        ROMCLK                  ;
                                1$:     MOV     #BIT9,R5        ;SET EXPECTED RESULTS
                                2$:     MOV     #BCC,(R0)       ;BCC OPR "CRC16"
                                        ROMCLK                  ;EXECUTE
                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT3+BIT1,(R0)
                                        ROMCLK                  ;XFR BCC REG TO DVRIC
                                        MOV     @DVRIC,R4       ;READ RESULTS
                                        MOV     #2.,R3          ;PREPARE
                                3$:     CLC                     ;TO
                                        ROL     R4              ;POSITION
                                        DEC     R3              ;RESULT
                                        BNE     3$              ;OF BCC OPERATION
                                        CMP     R5,R4           ;DID CRC16 WORK?
                                        BEQ     4$              ;BR IF YES
                                        HLT     6               ;INCORRECT BCC RESULTS
                                4$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
                                        ROMCLK                  ;ALU "F=A+B"
                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
                                        ROMCLK                  ;XFR ALU RESULT TO "B" REG
```

E08


```
                                        AUD     #BIT9,R5        ;UPDATE DATA COMPARE
                                        BIT     #BIT14,R5       ;ALL DATA DONE?
                                        BEQ     3$              ;BR IF NO
                                        SCOPE                   ;SCOPE TEST



                                        ;************************ TEST 105 ******************************
                                        ;*TEST OF THE BCC OPERATION USING
                                        ;*USING CRC16 FOR THE POLYNOMIAL
                                        ;*SPECIFIC DATA PATTERNS ARE USED TO
                                        ;*ISOLATE FAULTS AS SOON AS POSSIBLE
                                        ;;***************************************************************

                                ;  TEST 105
                                ;--------------
                                *ST105: MOV     #105,ISTNO
                                        MOV     #+TST105,NEXT
                                        MOV     #CRC16,XPOLY    ;SET POLYNOMIAL
                                        MOV     DVSEL,R0        ;SET REGISTER POINTER
                                        MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
                                                                ;SET FOR XFER FROM BCC TO DVRIC
                                        MOV     #1$,LOCK        ;SET IF SW09=1
                                1$:     MSTCLR                  ;ISSUE A MSTCLR
                                        MOV     #0,I3,SDVSCR    ;SET SOURCE SEL
                                        CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                        JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                        0.                      ;SHIFTS PERFORMED
                                        0                       ;DATA CHAR
                                        0                       ;PREVIOUS BCC
                                        JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                        0                       ;DATA TO THE A REGISTER
                                        0                       ;DATA TO THE B REGISTER
                                        10                      ;POLY SELT. (RAMOUTPUT)
                                        MOV     #BCC,(R0)
                                        ROMCLK                  ;DO A BCC CALCULATION (HROW)
                                        MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                        ROMCLK                  ;
                                        MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                        MOV     @DVRIC,R4       ;PUT HROW BCC INTO RESULTANT
                                        CMP     R5,R4           ;DOES EXPECTED = FOUND?
                                        BEQ     4$              ;BR IF OK!
                                        HLT     6               ;BCC CALCULATION ERROR
                                4$:     SCOP1                   ;SW09=1?
                                        MOV     #2$,LOCK        ;SET IF SW09=1
                                2$:     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
                                        0.                      ;SHIFTS PERFORMED
                                        ^B(10101010)            ;DATA CHAR
                                        @#TI14                  ;PREVIOUS BCC
                                        JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                        ^B(10101010)            ;DATA TO THE A REGISTER
                                        @#TI14                  ;DATA TO THE B REGISTER
                                        10                      ;POLY SELT. (RAMOUTPUT)
```

```
                                                MOV     #BCC,(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     DDVRIC,R4       ;PUT HRDW BCC INTO RECEIVED
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     65$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                        65$:    SCOP1                   ;SW09=1?
                                                MOV     #3$,LOOK        ;SET IF SW09=1
                                        3$:     MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,DDVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                0                       ;SHIFTS PERFORMED
                                                ^B<01010101>                    ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                ^B<01010101>                    ;DATA TO THE A REGISTER
                                                0                               ;DATA TO THE B REGISTER
                                                10                              ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     DDVRIC,R4       ;PUT HRDW BCC INTO RECEIVED
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     66$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                        66$:    SCOP1                   ;SW09=1?
                                                MOV     #4$,LOOK        ;SET IF SW09=1
                                        4$:     MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,DDVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                0                       ;SHIFTS PERFORMED
                                                ^B<11111111>                    ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                ^B<11111111>                    ;DATA TO THE A REGISTER
                                                0                               ;DATA TO THE B REGISTER
                                                10                              ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     DDVRIC,R4       ;PUT HRDW BCC INTO RECEIVED
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     67$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                        67$:    SCOP1                   ;SW09=1?
                                                MOV     #5$,LOOK        ;SET IF SW09=1
                                        5$:     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
                                                0.                      ;SHIFTS PERFORMED
                                                :B<00000001>                    ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                :B<00000001>                    ;DATA TO THE A REGISTER
                                                0                               ;DATA TO THE B REGISTER
                                                10                      ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK
                                                ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND?
                                                BEQ     6$              ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                                SCOP1                   ;SW09=1?
                                                MOV     #6$,LOCK        ;SET IF SW09=1
                                                MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                0.                      ;SHIFTS PERFORMED
                                                :B<00000100>                    ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                :B<00000100>                    ;DATA TO THE A REGISTER
                                                0                               ;DATA TO THE B REGISTER
                                                10                      ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK
                                                ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND?
                                                BEQ     69$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                                SCOP1                   ;SW09=1?
                                                MOV     #7$,LOCK        ;SET IF SW09=1
                                                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
                                                0.                      ;SHIFTS PERFORMED
                                                :B<00010000>                    ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L:DATA       ;GO AND LOAD DATA INTO THE DV11
                                                :B<00010000>                    ;DATA TO THE A REGISTER
                                                0                               ;DATA TO THE B REGISTER
                                                10                      ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
```

```
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     70$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                        70$:    SCOPI                   ;SW09=1?
                                                MOV     #8$,LOCK        ;SET IF SW09=1
                                        8$:     MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                8.                      ;SHIFTS PERFORMED
                                                'B^D1000000             ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                'B^D10000000            ;DATA TO THE A REGISTER
                                                0                       ;DATA TO THE B REGISTER
                                                10                      ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)       ;
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     71$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                        71$:    SCOPI                   ;SW09=1?
                                                MOV     #9$,LOCK        ;SET IF SW09=1
                                        9$:     MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
                                                JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                                                8.                      ;SHIFTS PERFORMED
                                                'B^D00000110            ;DATA CHAR
                                                0                       ;PREVIOUS BCC
                                                JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                                                'B^D00000110            ;DATA TO THE A REGISTER
                                                0                       ;DATA TO THE B REGISTER
                                                10                      ;POLY SELT. (RAMOUTPUT)
                                                MOV     #BCC,(R0)       ;
                                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                                                ROMCLK                  ;
                                                MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
                                                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
                                                BEQ     72$             ;BR IF OK!
                                                HLT     6               ;BCC CALCULATION ERROR
                                        72$:    SCOPI                   ;SW09=1?
                                                MOV     #100$,LOCK      ;SET IF SW09=1
                                        100$:   MSTCLR                  ;ISSUE A MSTCLR
                                                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                                                CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
```

```
                025620  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                025624  000010                          8.                      ;SHIFTS PERFORMED
                025626  000300                          ^B<01010000>                    ;DATA CHAR
                025630  000000                          0                       ;PREVIOUS BCC
                025632  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                025636  000300                          ^B<01010000>                    ;DATA TO THE A REGISTER
                025640  000000                          0                       ;DATA TO THE B REGISTER
                025642  000010                          10                      ;POLY SELT. (RAMOUTPUT)
                025644  012710  060000                  MOV     #BCC,(R0)
                025650  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                025652  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                025654  104415                          ROMCLK                  ;
                025656  013705  031652                  MOV     @#BCC,R5        ;PUT SOFTWARE BCC INTO EXPECTED
                025662  017704  153500                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                025666  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
                025670  001401                          BEQ     73$             ;BR IF OK!
                025672  104006                          HLT     6               ;BCC CALCULATION ERROR
                025674  104401                  73$:    SCOP1                   ;SW09=1?
                025676  012737  025704  001220          MOV     #101$,@#LOCK    ;SET IF SW09=1
                025704  104412                  101$:   MSTCLR                  ;ISSUE A MSTCLR
                025706  012777  000010  153446          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                025714  005037  031652                  CLR     @#BCC           ;ZERO CALCULATED BCC (SOFTWARE)
                025720  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                025724  000010                          8.                      ;SHIFTS PERFORMED
                025726  000330                          ^B<11010000>                    ;DATA CHAR
                025730  000000                          0                       ;PREVIOUS BCC
                025732  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                025736  000330                          ^B<11010000>                    ;DATA TO THE A REGISTER
                025740  000000                          0                       ;DATA TO THE B REGISTER
                025742  000010                          10                      ;POLY SELT. (RAMOUTPUT)
                025744  012710  060000                  MOV     #BCC,(R0)
                025750  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                025752  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                025754  104415                          ROMCLK                  ;
                025756  013705  031652                  MOV     @#BCC,R5        ;PUT SOFTWARE BCC INTO EXPECTED
                025762  017704  153400                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
                025766  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
                025770  001401                          BEQ     74$             ;BR IF OK!
                025772  104006                          HLT     6               ;BCC CALCULATION ERROR
                025774  104401                  74$:    SCOP1                   ;SW09=1?
                025776  012737  026004  001220          MOV     #102$,@#LOCK    ;SET IF SW09=1
                026004  104412                  102$:   MSTCLR                  ;ISSUE A MSTCLR
                026006  012777  000010  153346          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
                026014  005037  031652                  CLR     @#BCC           ;ZERO CALCULATED BCC (SOFTWARE)
                026020  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
                026024  000010                          8.                      ;SHIFTS PERFORMED
                026026  000300                          ^B<11000000>                    ;DATA CHAR
                026030  000000                          0                       ;PREVIOUS BCC
                026032  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
                026036  000300                          ^B<11000000>                    ;DATA TO THE A REGISTER
                026040  000000                          0                       ;DATA TO THE B REGISTER
                026042  000010                          10                      ;POLY SELT. (RAMOUTPUT)
                026044  012710  060000                  MOV     #BCC,(R0)
                026050  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
                026052  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
                026054  104415                          ROMCLK                  ;
```

```
026056  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
026062  017704  153300                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
026066  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
026070  001401                          BEQ     75$             ;BR IF OK!
026072  104006                          HLT     6               ;BCC CALCULATION ERROR
026074  104401                  75$:    SCOPE1                  ;SW09=1?
026076  104400                          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     #TST107,NEXT
026114  012737  000200  031646          MOV     #LRC8,XPOLY     ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
026122  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
026126  013700  001400                  MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DVRIC
026132  005001                          CLR     R1              ;SET DATA POINTER TO 0
026134  005037  031652                  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  031652  026204          MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
026166  013737  031652  026214          MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
026174  004537  031474                  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  004337  031420                  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  031652                  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
                                ;--------------
001226  026256  012737  000107  001226  TST107: MOV     #107,TSTNO
001216  026264  012737  026434  001216          MOV     #TST110,NEXT
031646  026272  012737  120001  031646          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
030346  026300  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
001400  026304  013700  001400                  MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DVRIC
        026310  005001                          CLR     R1              ;SET DATA POINTER TO 0
031652  026312  005037  031652                  CLR     CALBCC          ;ZERO SOFTWARE BCC
        026316  104412                  1$:     MSTCLR                  ;CLEAR THE DV11
153034  026320  012777  000010  153034          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
026360  026326  010137  026360                  MOV     R1,2$           ;LOAD SOFTWARE CHAR
026370  026332  010137  026370                  MOV     R1,4$           ;LOAD HRDW CHAR
026362  026336  013737  031652  026362          MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
026372  026344  013737  031652  026372          MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
031474  026352  004537  031474                  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
031420  026364  004337  031420                  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
060000  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                  ;
031652  026410  013705  031652                  MOV     CALBCC,R5       ;GET GOOD BCC
152746  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$:     INCB    R1              ;UPDATE DATA CHAR
        026430  001332                          BNE     1$              ;BR IF NOT ALL DATA DONE.
        026432  104400                          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
                                ;--------------
001226  026434  012737  000110  001226  TST110: MOV     #110,TSTNO
001216  026442  012737  026612  001216          MOV     #TST111,NEXT
031646  026450  012737  102010  031646          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 DVRIC
 5315   026466   005001                               CLR     R1              ;SET DATA POINTER TO 0
 5316   026470   005037   031652                      CLR     CALBCC          ;ZERO SOFTWARE BCC
 5317   026474   104412                      1$:     MSTCLR                  ;CLEAR THE DV11
 5318   026476   012777   000010   152656             MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5319   026504   010137   026536                      MOV     R1,2$           ;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,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
 5324   026534   000010                               8.                      ;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
 5343
```

```
                                        ;********************** 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
```

```
 5400  027020  005037  177776                    CLR     PS              ;ZERO CPU PRIO.
 5401  027024  012777  010000  152330            MOV     #BIT12,@DVSCR   ;SET AN INTERUPT RELATIVE BIT.
 5402  027032  000240                            NOP                     ;WAIST TIME
 5403  027034  012777  002000  152320            MOV     #BIT10,@DVSCR   ;SET THE ALTERNATE RELATIVE BIT.
 5404  027042  000240                            NOP                     ;WAIST
 5405  027044  005077  152312                    CLR     @DVSCR          ;ZERO REG
 5406  027050  004537  031654                    JSR     R5,SETVEC       ;GO RESET VECTORS "A" AND "B"
 5407  027054  027126                            3$                      ;"A"
 5408  027056  027114                            2$                      ;"B"
 5409  027060     340     340                    .BYTE   340,340         ;PRIO. AT 7
 5410  027062  052777  012000  152272            BIS     #BIT12!BIT10,@DVSCR
 5411  027070  000240                            NOP                     ;SET BOTH INTERUPT RELATIVE BITS.
 5412  027072  104010                            HLT     10              ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
 5413  027074  004537  031654          1$:       JSR     R5,SETVEC       ;RESET VECTORS
 5414  027100  031676                            NO.ATRAP                ;"A"
 5415  027102  031702                            NO.BTRAP                ;"B"
 5416  027104     340     340                    .BYTE   340,340
 5417  027106  005077  152250                    CLR     @DVSCR          ;DSABLE DV11
 5418  027112  104400                            SCOPE                   ;SCOPE THIS TEST.
 5419  027114  012706  001200          2$:       MOV     #STACK,SP       ;RESET STACK
 5420  027120  005077  152236                    CLR     @DVSCR          ;DSABLE DV11
 5421  027124  000763                            BR      1$              ;RETURN
 5422  027126  104011                3$:       HLT     11              ;VECTOR HERE WAS WRONG SIDE
 5423  027130  000771                            BR      2$
 5424
 5425                                    ;************************ TEST 113 ******************************
 5426                                    ;*TEST THAT SETTING BIT15!BIT9 AND BIT13!BIT9
 5427                                    ;*NPR STAT INTR AND NPR STAT IE PRODUCE AN INTERUPT ON VECTOR "3"
 5428                                    ;:****************************************************************
 5429
 5430                                    ;  TEST 113
 5431                                    ;--------------
 5432  027132  012737  000113  001226  TST113: MOV     #113,TSTNO
 5433  027140  012737  027310  001216          MOV     #TST114,NEXT
 5434  027146  012737  000340  177776          MOV     #340,PS         ;LOCK OUT CPU INTERUPTS
 5435  027154  104412                          MSTCLR                  ;ISSUE DVRESET
 5436  027156  004537  031654                  JSR     R5,SETVEC       ;GO SET VECTOR "A" AND "B"
 5437  027162  031676                          NO.ATRAP                ;"A"
 5438  027164  031702                          NO.BTRAP                ;"B"
 5439  027166     340     340                  .BYTE   340,340         ;PRIO. AT 7
 5440  027170  005037  177776                  CLR     PS              ;ZERO CPU PRIO.
 5441  027174  012777  101000  152160          MOV     #BIT15!BIT9,@DVSCR      ;SET AN INTERUPT RELATIVE BIT.
 5442  027202  000240                          NOP                     ;WAIST TIME
 5443  027204  012777  021000  152150          MOV     #BIT13!BIT9,@DVSCR      ;SET THE ALTERNATE RELATIVE BIT.
 5444  027212  000240                          NOP                     ;WAIST
 5445  027214  005077  152142                  CLR     @DVSCR          ;ZERO REG
 5446  027220  012777  001000  152134          MOV     #BIT9,@DVSCR    ;SET SYS MAINT ENABLE
 5447  027226  004537  031654                  JSR     R5,SETVEC       ;GO RESET VECTORS "A" AND "B"
 5448  027232  027304                          3$                      ;"A"
 5449  027234  027272                          2$                      ;"B"
 5450  027236     340     340                  .BYTE   340,340         ;PRIO. AT 7
 5451  027240  052777  121000  152114          BIS     #BIT15!BIT9!BIT13!BIT9,@DVSCR
 5452  027246  000240                          NOP                     ;SET BOTH INTERUPT RELATIVE BITS.
 5453  027250  104010                          HLT     10              ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
 5454  027252  004537  031654          1$:     JSR     R5,SETVEC       ;RESET VECTORS
 5455  027256  031676                          NO.ATRAP                ;"A"
```

```
                                        NU.BTRAP                ;"B"
                                .BYTE   340,340
                                CLR     @DVSCR                  ;DSABLE DV11
                                SCOPE                           ;SCOPE THIS TEST.
                        2$:     MOV     #STACK,SP               ;RESET STACK
                                CLR     @DVSCR                  ;DSABLE DV11
                                SEC                             ;RETURN
                        3$:     RTT                             ;VECTOR HERE WAS WRONG SIDE
                                NOP

                        ;************************ 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     #TST115,NEXT
                                MOV     #340,PS                 ;SET CPU TO PRIO 7
                                MSICLR                          ;RESET DV11
                                CLR     R3                      ;SET COUNTER TO 0
                                MOV     #3,R0                   ;SET TO GET 2 INTERRUPTS
                                JSR     R5,SETVEC               ;SET DV11 VECTORS
                                .WORD                           ;RECEIVER INTERRUPTS TO 2$
                                .WORD                           ;TRANSMITTER INTERRUPTS TO 3$
                                .BYTE   340,340                 ;SET PRIORITY TO 7 ON INTERRUPT
                                MOV     #BIT4,@DVSCR            ;SET SYSTEM MAINT
                                BIS     #BIT15+BIT13,@DVSCR
                                BIS     #BIT9+BIT7+BIT6,@DVSCR
                                CLR     @DVSCR                  ;SET B INTERRUPT ;A INTERRUPT ;CLEAR CPU STATUS
                                INCB    R3                      ;HANG WAITING FOR INTERRUPTS
                                BNE     .-2                     ;
                                TST     R0                      ;DID TWO INTERRUPTS OCCUR?
                                BPL     4$                      ;BR IF YES
                                CLR     R0                      ;EITHER NOT ENOUGH(2) OR TOO MANY (72) INTERRUPTS
                        1$:     SCOPE                           ;SCOPE TEST
                        2$:     TSTB    @RCSR                   ;RXISR
                                CMPB    #1,R0                   ;IF RX GOT HERE 1ST R0-1 S B=1
                                BEQ     6$                      ;BR IF OK
                                INC     R0                      ;RX NOT 1ST
                        6$:     MOV     #BIT7+BIT6,@DVSCR       ;DISQUALIFY INTERRUPT REQST
                                RTI                             ;RETURN
                        3$:     TSTB    @XCSR                   ;TX ISR
                                BEQ     4$                      ;IF SCOND INTERRUPT R0 SHOULD NOW =0
                                INC     R0                      ;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
                                        ;--------------
027450  012737  000115  001686  TST115: MOV     #115,TSTNO
027456  012737  027546  001672          MOV     #TST116,NEXT
027464  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERUPTS
027472  104413                          MSTCLR                          ;CLEAN DV11
027474  004737  031654                  JSR     R5,SETVEC               ;PREPARE VECTORS
027500  031676                          NO.ATRAP                                ;"A"
027502  031706                          NO.BTRAP                        ;"B"
027504      340                 340     .BYTE   340,340                 ;PRIO, AT 7
027506  012737  001000  001346          MOV     #DI+BIT7+BIT6,DVSCR
027514  012737  000340  177776          MOV     #340,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027522  000240                          NOP                             ;WAIST
027524  005037  151632          1$:     CLR     DVSCR                   ;DISABLE DV11
027530  104403                          SCOPE                           ;SCOPE THIS TEST
027532  012716  027614          2$:     MOV     #1$,(SP)                ;SET FOR RETURN
027536  000002                          RTI                             ;

                                        ;********************** TEST 116 ****************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 5 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;;***********************************************************

                                        ;  TEST 116
                                        ;--------------
027540  012737  000116  001686  TST116: MOV     #116,TSTNO
027546  012737  027636  001672          MOV     #TST117,NEXT
027554  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERUPTS
027562  104413                          MSTCLR                          ;CLEAN DV11
027564  004737  031654                  JSR     R5,SETVEC               ;PREPARE VECTORS
027570  031676                          NO.ATRAP                                ;"A"
027572  031706                          NO.BTRAP                        ;"B"
027574      340                 340     .BYTE   340,340                 ;PRIO, AT 7
027576  012777  001000  151542          MOV     #DI+BIT7+BIT6,@DVSCR
027604  012737  000300  177776          MOV     #300,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027612  000240                          NOP                             ;WAIST
027614  005037  151542          1$:     CLR     @DVSCR                  ;DISABLE DV11
027620  104403                          SCOPE                           ;SCOPE THIS TEST
027622  012716  027614          2$:     MOV     #1$,(SP)                ;SET FOR RETURN
027626  000002                          RTI                             ;

                                        ;********************** TEST 117 ****************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 5 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;;***********************************************************

                                        ;  TEST 117
                                        ;--------------
027630  012737  000117  001686  TST117: MOV     #117,TSTNO
027636  012737  027726  001672          MOV     #TST120,NEXT
027644  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERUPTS
```

```
027666  104412                  MSTCLR                  ;CLEAN DV11
027670  004537  031654           JSR     R5,SETVEC       ;PREPARE VECTORS
027674  031660                  NO.ATRAP                        ;"A"
027676  031702                  NO.BTRAP                ;"B"
027700      340                 .BYTE   340,340         ;PRIO. AT 7
027701  012737  001300  151466   MOV     #BIT9+BIT7+BIT6,@DVSCR
027707  012737  000340  177776   MOV     #240,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027715  000240                  NOP                     ;WAIST
027716  005037  151462      1$: CLR     @DVSCR          ;DISABLE DV11
027722  104000                  SCOPE                   ;SCOPE THIS TEST
027724  012716  027704      2$: MOV     #1$,(SP)        ;SET FOR RETURN
027730  000002                  RTI                     ;


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

                                ;  TEST 120
                                ;--------------
027732  012737  000120  001226  TST120: MOV     #120,TSTNO
027740  012737  027760  001216          MOV     #TST121,NEXT
027746  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
027754  104412                  MSTCLR                  ;CLEAN DV11
027756  004537  031654           JSR     R5,SETVEC       ;PREPARE VECTORS
027762  000004                  4                       ;"A"
027764  031702                  NO.BTRAP                ;"B"
027766      340                 .BYTE   340,340         ;PRIO. AT 7
027767  012737  001300  151376          MOV     #BIT9+BIT7+BIT6,@DVSCR
027775  012737  000200  177776          MOV     #200,PS ;SET CPU PRIO AND ENABLE VECT. "A"
030003  000240                  NOP                     ;WAIST
030004  104007                  CLR     -(PC)           ;DV FAILED TO INTERUPT
030006  005077  151360      1$: CLR     @DVSCR          ;DISABLE DV11
030012  104400                  SCOPE                   ;SCOPE THIS TEST
030014  012716  027776      2$: MOV     #1$,(SP)        ;SET FOR RETURN
030020  000002                  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 BITS.
                                ;;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                ;  TEST 121
                                ;--------------
030022  012737  000121  001226  TST121: MOV     #121,TSTNO
030030  012737  030114  001216          MOV     #TST122,NEXT
030036  104412                  MSTCLR                  ;RESET DV11
030040  005777  151326          TST     @DVSCR          ;MAKE SURE 15 =0
030044  100001                  BPL     64$             ;BR IF OK
030046  104000                  HLT     0               ;15 = 1 B=0
030050  012777  000010  151314  64$:    MOV     #BIT3,@DVSCR    ;SET SOURCE SEC
```

E09


```
                    030046  016777  030367  151324          MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT3+BIT1+BIT0,@DVSFR
                    030054  005000                          CLR     R0                      ;XFR ALU RESULT TO DVNSR
                    030056  104416                          ROMCLK                          ;EXECUTE
                    030060  005767  151276          1$:     TST     @DVSCR                  ;15=1?
                    030064  100404                          BMI     2$                      ;BR IF YES
                    030066  104414                          DELAY                           ;WASTE TIME
                    030070  005760  000100                  INC     R0                      ;DO DELAY AND WAIT
                    030074  001372                          BNE     1$                      ;
                    030076  104000                          TRAP    0                       ;IS NEVER SET
                    030100  005767  151300          2$:     TST     @DVNSR                  ;READ NSR
                    030104  005767  151264                  TST     @DVSCR                  ;DID IS CLEAR IN SCR?
                    030110  100001                          BPL     3$                      ;BR IF YES
                    030112  104000                          TRAP    0                       ;DVSCR ID NOT CLEARED BY READING NSR
                    030114  104425                  3$:     SCOPE                           ;SCOPE TEST

                                                    ;+++++++++++++++++++++++ TEST 122 ++++++++++++++++++++++++++++
                                                    ;+TEST TO WRITE PATTERNS THROUGH
                                                    ;+THE NPR STATUS REGISTER.
                                                    ;+BITS WRITTEN: 11,10,09,08,03,02,01,00
                                                    ;+(WHEN BIT IS OF DVSCR IS SET SO SHOULD BIT IS OF DVNSR)
                                                    ;;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                                    ;  TEST 122
                                                    ;--------------
                    030116  012737  000122  001000  TST122: MOV     #122,ISTNO
                    030124  012746  030134  000010          MOV     #TST120,NEXT
                    030132  012706  030172  001200          MOV     #1$,LOCK
                    030136  104416                          MSTCLR                          ;RESET DV11
                    030140  012777  000010  151214          MOV     #BIT3,@DVSCR            ;SET SOURCE SE
                    030146  005000                          CLR     R0                      ;ZERO LINE NUMBER POINTER
                    030150  005037  001250                  CLR     TEMP2                   ;ZERO DATA
                    030154  012703  001400                  MOV     #DVNSR,R3               ;SET POINTER
                    030160  012702  001400                  MOV     #DVSFR,R2               ;
                    030164  012737  100000  001246          MOV     #BIT15,TEMP1            ;SET CORRESPONDING BIT TO BE FOUND IN NSR.
                    030172  012705  030267          1$:     MOV     #XFR+BIT7+BIT5+BIT4+BIT3+BIT1+BIT0,R5
                    030176  104416                          ROMCLK                          ;XFR RAM OUTPUT TO NSR
                    030200  005767  151156                  TST     @DVSCR                  ;WAIT FOR
                    030204  100375                          BPL     .-4                     ;
                    030206  013700  001246                  MOV     TEMP1,R0                ;GET GOOD 15=1 DATA
                    030212  011304                          MOV     (R3),R4                 ;READ NSR
                    030214  020004                          CMP     R0,R4                   ;OK?
                    030216  001401                          BEQ     2$                      ;BR IF GOOD
                    030220  104413                          HALT    13                      ;DVNSR HAS WRONG DATA
                    030222  104401                  2$:     SCOPE                           ;LOOK ON DATA?
                    030224  004537  031342                  PERFORM .SETSCAN                ;UPDATE MSCANNER
                    030230  000001                          .                               ;BY ONE
                    030232  005237  001246                  INC     TEMP1                   ;UPDATE DATA
                    030236  032700  000020                  BIT     #BIT4,R0                ;ALL DONE?
                    030242  001253                          BNE     1$                      ;BR IF NO.
                    030244  042737  077777  001246          BIC     #^C<BIT15>,TEMP1        ;CLEAR ALL BUT ENTRY PRESENT
                    030252  005000                          CLR     R0                      ;ZERO LINE POINTER
                    030254  005001                          CLR     R1                      ;ZERO MSCANNER, POINTER
                    030256  005237  001250                  INC     TEMP2                   ;UPDATE HIGH BYTE DATA
                    030262  153737  001250  001247          BISB    TEMP2,TEMP1+1           ;PLACE IN GOOD LOCATION
                    030270  010212                          MOV     TEMP2,(R2)              ;RAM ADDRESS
```

```
DZDVA-B MACY11 27(732)  17-SEP-76  14:02  PAGE 110
DZDVAB.P11      DV11 DEVICE DIAGNOSTICS.    COPYRIGHT 1975  DIGITAL EQUIP. CORP.

                                            BIS     #RAM,(R2)       ;RAM READ
                                            ROMCLK                  ;
                                            BIT     #BIT4,TEMP2     ;ALL DONE?
                                            BEQ     1$              ;BR IF NO
                                            SCOPE                   ;SCOPE TEST.

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

                                            ;************************ TEST 123 ****************************
                                            ;*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 DV11 AND XFERED
                                            ;*INTO THE DVRIC REGISTER.
                                            ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
                                            ;;*************************************************************

                                        ;  TEST 123
                                        ;--------------
030314  012737  000123  001226  TST123: MOV     #123,TSTNO
030320  012737  030464  001216          MOV     #TST124,NEXT
030330  012737  030412  001220          MOV     #1$,LOCK
030336  104412                          MSTCLR                  ;RESET DV11
030340  012777  000010  151014          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
030346  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
030352  013702  001400                  MOV     DVSFR,R2        ;
030356  012777  032756  151012          MOV     #NPRLOC,@DVSRA
030364  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
030370  104414                          ROMCLK                  ;EXECUTE
030372  012712  030263                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
030376  104414                          ROMCLK                  ;DO XFR TO NPR ADD.
030400  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
030404  005006                          CLR     R5              ;CLEAR GOOD
030406  005077  151764                  CLR     @DVSRA          ;CLEAR DATA PORT
030412  012712  040000          1$:     MOV     #NPR,(R2)       ;DO THE NPR
030416  104414                          ROMCLK                  ;***NOW***
030420  012712  030226                  MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
030424  104414                          ROMCLK                  ;XFR NPR OUTPUT TO D.RIC
030430  011304                          MOV     (R3),R4         ;READ RIC
030432  013705  032756                  MOV     NPRLOC,R5 ;READ GOOD DATA
030436  020504                          CMP     R5,R4           ;OK?
030440  001401                          BEQ     2$              ;BR IF YES
030442  104413                          HLT     13              ;NPR FUNCTION FAILED
030444  104401          2$:     SCOPE1                          ;LOOP?
030446  062737  032756                  INCB    NPRLOC ;UPDATE DATA
030452  001360                          BNE     1$              ;BR IF MORE
030454  104400                          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 DV11 AND XFERED
                                        ;*INTO THE DVRIC REGISTER.
                                        ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
                                        ;*****************************************************************

                                ;  TEST 124
                                ;--------------
                                TST124: MOV     #124,TSTNO
                                        MOV     #TST126,.NEXT
                                        MOV     #1$,.LOOP
                                        MSTCLR                  ;RESET DV11
                                        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
                                        MOV     DVRIC,R3        ;SET POINTERS
                                        MOV     DVSFR,R2        ;
                                        MOV     #NPRLOC+1,@DVSRA
                                        MOV     #RAM,(R2)       ;RAM READ
                                        ROMCLK                  ;EXECUTE
                                        MOV     #XFR+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
                                        ROMCLK                  ;DO XFR TO NPR ADD.
                                        CLR     NPRLOC          ;CLEAR NPR LOCATION
                                        CLR     R5              ;CLEAR GOOD
                                        CLR     @DVSRA          ;CLEAR DATA PORT
                                1$:     MOV     #NPR,(R2)       ;DO THE NPR
                                        ROMCLK                  ;***NOW***
                                        MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
                                        ROMCLK                  ;XFR NPR OUTPUT TO DVRIC
                                        MOV     (R3),R4         ;READ RIC
                                        MOV     NPRLOC,R5       ;GET GOOD DATA
                                        SWAB    R5              ;
                                        CMP     R5,R4           ;OK?
                                        BEQ     2$              ;BR IF YES
                                        HLT     13              ;NPR FUNCTION FAILED
                                2$:     SCOPE                   ;LOOP.
                                        INCB    NPRLOC+1 ;UPDATE DATA
                                        BNE     1$              ;BR IF MORE
                                        SCOPE                   ;SCOPE TEST



                                        ;************************ TEST 125 ******************************
                                        ;*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 EVEN ADDRESS FOR THE NPR OPERATION
                                        ;*****************************************************************

                                ;  TEST 125
                                ;--------------
                                TST125: MOV     #125,TSTNO
                                        MOV     #TST126,.NEXT
                                        MOV     #1$,.LOOP
                                        MSTCLR                  ;RESET DV11
```

```
031040  012777  000010  150512          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
031046  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
031052  013702  001400                  MOV     DVSFR,R2        ;
031056  012777  032756  150510          MOV     #NPRLOC,@DVSRA
031064  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
031070  104414                          ROMCLK                  ;EXECUTE
031072  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
031076  104414                          ROMCLK                  ;DO XFR TO NPR ADD.
031100  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
031104  005005                          CLR     R5              ;CLEAR GOOD
031106  005037  150462                  CLR     @DVSRA          ;CLEAR DATA PORT
031112  005037  032756          10:     CLR     NPRLOC          ;CLEAR NPR LOC
031116  012712  020000                  MOV     #RAM,(R2)       ;DO RAM READ
031122  104414                          ROMCLK                  ;
031124  012712  030265                  MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
031130  104414                          ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
031132  012712  040000                  MOV     #NPR,(R2)       ;DO THE NPR
031136  104414                          ROMCLK                  ;***NOW***
031140  017705  150430                  MOV     @DVSRA,R5       ;READ GOOD DATA
031144  013704  032756                  MOV     NPRLOC,R4 ;READ ???? DATA
031150  020405                          CMP     R5,R4           ;OK?
031152  001403                          BEQ     20$             ;BR IF YES
031154  104401                          HALT    10              ;NPR FUNCTION FAILED
031156  005304                  20$:    SOOP1                   ;LOOP?
031160  001410                          BNE     30$             ;UPDATE DATA
031162  032777  000400  150402          BIT     #BIT8,@DVSRA    ;ALL DONE?
031170  001346                          BNE     10$             ;BR IF NO
031172  104400                          SCOPE                   ;SCOPE TEST
```

```
                        ;*********************** TEST 126 ****************************
                        ;*BASIC TEST OF I ' 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
                    ;--------------
031000  012737  000126  001226  TST126: MOV     #126,TSTNO
031006  012737  031154  001216          MOV     #TST127,NEXT
031014  012737  031076  001220          MOV     #1$,LOOK
031022  104412                          MSTCLR                  ;RESET DV11
031024  012777  000010  150330          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
031032  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
031036  013702  001400                  MOV     DVSFR,R2        ;
031042  012777  032757  150326          MOV     #NPRLOC+1,@DVSRA
031050  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
031054  104414                          ROMCLK                  ;EXECUTE
031056  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
031062  104414                          ROMCLK                  ;DO XFR TO NPR ADD.
031064  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
031070  005005                          CLR     R5              ;CLEAR GOOD
```

```
031072  005077  150300          CLR     @DVSRA          ;CLEAR DATA PORT
031076  005037  032756          CLR     NPRLOC          ;CLEAR NPR LOC
031102  012712  020000          MOV     #RAM,(R2)       ;DO RAM READ
031106  104415                  ROMCLK                  ;
031110  012712  030265          MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
031114  104415                  ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
031116  012712  040000          MOV     #NPR,(R2)       ;DO THE NPR
031122  104415                  ROMCLK                  ;***NOW***
031124  017705  150246          MOV     @DVSRA,R5       ;READ GOOD DATA
031130  013704  032756          MOV     NPRLOC,R4       ;GET DATA
031134  000304                  SWAB    R4              ;
031136  020504                  CMP     R5,R4           ;OK?
031140  001401                  BEQ     2$              ;BR IF YES
031142  104013                  HLT     13              ;NPR FUNCTION FAILED
031144  104404          3$:     SCOPE                   ;LOOP?
031146  005277  150224          INC     @DVSRA          ;UPDATE DATA
031152  032777  000400  150216  BIT     #BIT8,@DVSRA    ;ALL DONE?
031160  001746                  BEQ     1$              ;BR IF NO
031162  104400                  SCOPE                   ;SCOPE TEST
```

```
;************************ TEST 127 ******************************
;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
;*TEST THAT THE DV11 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
;--------------
031164  012737  000127  001226  TST127: MOV     #127,TSTNO
031172  012737  002436  001216          MOV     #.EOP,NEXT
031200  104412                  MSTCLR                  ;RESET DV11
031202  012777  000010  150152          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
031210  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
031214  013702  001400                  MOV     DVSFR,R2        ;
031220  052777  000060  150134          BIS     #BIT5+BIT4,@DVSCR       ;SET EA BITS.
031226  012777  177320  150142          MOV     #177320,@DVSRA  ;LOAD "NXM".
031234  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
031240  104415                  ROMCLK                  ;EXECUTE
031242  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
031246  104415                  ROMCLK                  ;EXECUTE
031250  012712  040000                  MOV     #NPR,(R2)       ;DO THE NPR
031254  104415                  ROMCLK                  ;***NOW***
031256  012712  003000                  MOV     #BIT10+BIT9,(R2)        ;BRANCH "A" NXM
031262  017704  150102                  MOV     @DVLCR,R4       ;READ BRANCH TEST POINTS.
031266  010405                  MOV     R4,R5           ;GET IMAGE
031270  042705  000001                  BIC     #BIT0,R5        ;BR "A" TRUE
031274  052705  000002                  BIS     #BIT1,R5        ;BR B FALSE
031300  020504                  CMP     R5,R4           ;NXM TRUE?
031302  001401                  BEQ     .+4             ;BR IF YES
031304  104006                  HLT     6               ;BR "A" OF NXM FAILED!
031306  012712  030017                  MOV     #S.0+BIT3+BIT2+BIT1+BIT0,(R2)
```

J09


```
                                                                ;SET/CLEAR  CLEAR NXM.
031312  104415                          ROMCLK                  ;EXECUTE.
031314  012712  003000                  MOV     #BIT10+BIT9,(R2)        ;BRANCH "A" NXM
031320  017704  150044                  MOV     @DVLCR,R4       ;READ BRANCH TEST POINTS.
031324  010405                          MOV     R4,R5           ;GET IMAGE
031326  042705  000003                  BIS     #BIT1+BIT0,R5   ;BR A FALSE  BR B FALSE
031332  020504                          CMP     R5,R4           ;NXM TRUE?
031334  001403                          BEQ     .+4             ;BR IF NO
031336  104000                          HLT     6               ;BR "A" OF NXM FAILED TO CLEAR!
031340  104400                          SCOPE                   ;SCOPE TEST
```

```
5915
5916  031342                                SETSCAN:
5917  031342  010346                              MOV     R3,-(SP)
5918  031344  052777  000010  150010              BIS     #BIT3,@DVSCR
5919  031352  012503                              MOV     (R5)+,R3
5920  031354  001414                              BEQ     2$
5921  031356  012777  050102  150014  1$:         MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
5922  031364  104415                              ROMCLK
5923  031366  005201                              INC     R1
5924  031370  012777  050102  150002              MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
5925  031376  104415                              ROMCLK
5926  031400  005201                              INC     R1
5927  031402  005303                              DEC     R3
5928  031404  001364                              BNE     1$
5929  031406  012603                      2$:     MOV     (SP)+,R3
5930  031410  010100                              MOV     R1,R0
5931  031412  000241                              CLC
5932  031414  006000                              ROR     R0
5933  031416  000205                              EXIT
5934
5935                                            ;SUBROUTINE TO LOAD DATA INTO "A" AND "B" REG.
5936                                            ;THE FIRST ARGUMENT WILL LOAD THE "A" REGISTER;
5937                                            ;THE SECOND ARGUMENT WILL LOAD THE "B" REGISTER;
5938                                            ;AND THE THIRD ARGUMENT WILL SPECIFY THE POLYNOMIAL USED.
5939
5940  031420  012377  147752          L.DATA: MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE "A" REG
5941  031424  012710  020000                  MOV     #RAM,(R0)       ;DO A ROM READ INSTR.
5942  031430  104415                          ROMCLK                  ;EXECUTE
5943  031432  012710  030261                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
5944  031436  104415                          ROMCLK                  ;DATAXFR FROM RAM OUTPUT TO "A" REGISTER
5945  031440  012377  147732                  MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE "B" REG.
5946  031444  012710  020000                  MOV     #RAM,(R0)       ;DO A ROM READ
5947  031450  104415                          ROMCLK                  ;
5948  031452  012710  030262                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
5949  031456  104415                          ROMCLK                  ;DO A DATA XFER FROM RAM OUTPUT TO THE "B" REG.
5950  031460  012377  147712                  MOV     (R3)+,@DVSRA    ;PLACE DATA TO REMAIN IN THE RAMOUTPUT REG
5951  031464  012710  020000                  MOV     #RAM,(R0)       ;DO A RAM READ TO SPECIFY POLYN.
5952  031470  104415                          ROMCLK                  ;READ
5953  031472  000203                          RTS     R3              ;LEAVE HERE
5954
5955
5956  031474  010046                  SIMBCC: MOV     R0,-(SP)
5957  031476  010146                          MOV     R1,-(SP)
5958  031500  010245                          MOV     R2,-(SP)
5959  031502  012537  001246                  MOV     (R5)+,TEMP1
5960  031506  012537  001250                  MOV     (R5)+,TEMP2
5961  031512  012537  001252                  MOV     (R5)+,TEMP3
5962  031516  005037  031650          1$:     CLR     BCCFBK
5963  031522  013700  001252                  MOV     TEMP3,R0
5964  031526  006037  001250                  ROR     TEMP2
5965  031532  005500                          ADC     R0
5966  031534  032700  000001                  BIT     #BIT0,R0
5967  031540  001402                          BEQ     2$
5968  031542  005137  031650                  COM     BCCFBK
5969  031546  013700  031646          2$:     MOV     XPOLY,R0
5970  031552  005100                          COM     R0
```

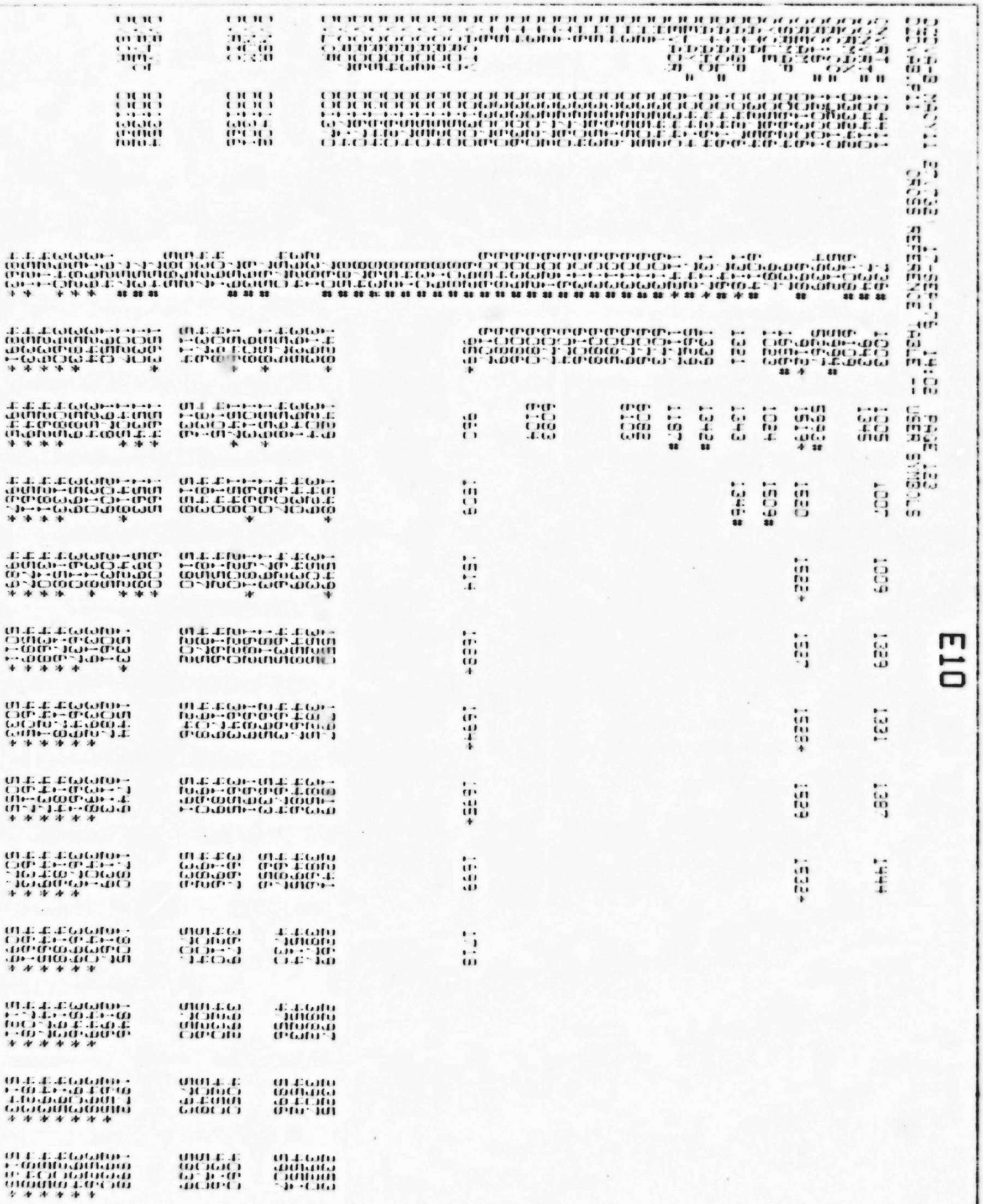
```
 5971  031554  040037  031650                  BIC     R0,BCCFBK
 5972  031560  000241                          CLC
 5973  031562  006037  001252                  ROR     TEMP3
 5974  031566  013700  031650                  MOV     BCCFBK,R0
 5975  031572  013701  001252                  MOV     TEMP3,R1
 5976  031576  010102                          MOV     R1,R2
 5977  031600  040100                          BIC     R1,R0
 5978  031602  043702  031650                  BIC     BCCFBK,R2
 5979  031606  050200                          BIS     R2,R0
 5980  031610  043737  031646  001252          BIC     XPOLY,TEMP3
 5981  031616  050037  001252                  BIS     R0,TEMP3
 5982  031622  005337  001246                  DEC     TEMP1
 5983  031626  001363                          BNE     1$
 5984  031630  013737  001252  031652          MOV     TEMP3,CALBCC
 5985  031636  012602                          MOV     (SP)+,R2
 5986  031640  012601                          MOV     (SP)+,R1
 5987  031642  012600                          MOV     (SP)+,R0
 5988  031644  000205                          RTS     R5
 5989  031646  000000              XPOLY:  0
 5990  031650  000000              BCCFBK: 0
 5991  031652  000000              CALBCC: 0
 5992          000200              LRC8=200
 5993          120001              CRC16=120001
 5994          102010              CRC.CCITT=102010
 5995
 5996
 5997  031654                      SETVEC:
 5998  031654  012577  147472              MOV     (R5)+,@DVRVEC
 5999  031660  012577  147472              MOV     (R5)+,@DVTVEC
 6000  031664  112577  147464              MOVB    (R5)+,@DVRLVL
 6001  031670  112577  147464              MOVB    (R5)+,@DVTLVL
 6002  031674  000205                      RTS     R5
 6003
 6004  031676                      NO.ATRAP:
 6005  031676  104011                      HLT     11
 6006  031700  000002                      RTI
 6007
 6008  031702                      NO.BTRAP:
 6009  031702  104012                      HLT     12
 6010  031704  000002                      RTI
 6011
 6012
```

```
6013                                            .NLIST  BEX
        031706  050377  044522  040515  EM1:    .ASCIZ  <377>/PRIMARY REGISTER ADDRESSING TIME-OUT/
        031754  051777  041505  047117  EM2:    .ASCIZ  <377>↑SECONDARY REGISTER READ/WRITE TEST↑
        032020  050377  044522  040515  EM3:    .ASCIZ  <377>↑PRIMARY REGISTER READ/WRITE TEST↑
        032062  046777  046505  051117  EM4:    .ASCIZ  <377>↑MEMORY EXTENSION READ/WRITE TEST↑
        032124  051777  042520  044503  EM5:    .ASCIZ  <377>/SPECIAL FUNCTION REG TEST/
        032157     377  042526  052103  EM6:    .ASCIZ  <377>/VECTOR "A" FAILED TO INTERUPT/
        032216  053377  041505  047524  EM7:    .ASCIZ  <377>/VECTOR "B" FAILED TO INTERUPT/
        032255     377  047125  054105  EM8:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "A"/
        032320  052777  042516  050130  EM9:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "B"/
        032363     377  051120  046511  EM10:   .ASCIZ  <377>/PRIMARY REGISTER ERROR/
        032413     377  044514  042516  EM11:   .ASCIZ  <377>/LINE CARD STATIC TEST/
        032442  051377  043505  051511  DH1:    .ASCIZ  <377>/REGISTER REFERENCED  TRAPPED FROM/
        032505     377  054105  042520  DH2:    .ASCIZ  <377>/EXPECTED  FOUND   LINE  SEC REG  PRI REG/
        032557     377  054105  042520  DH3:    .ASCIZ  <377>/EXPECTED  FOUND   PRI REG/
        032612  045377  051123  050040  DH4:    .ASCIZ  <377>/JSR PC  DVSFR   EXPECTED  FOUND/
        032663     377  053104  043123  DH5:    .ASCIZ  <377>/DVSFR   EXPECTED  FOUND/
        032704  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
```

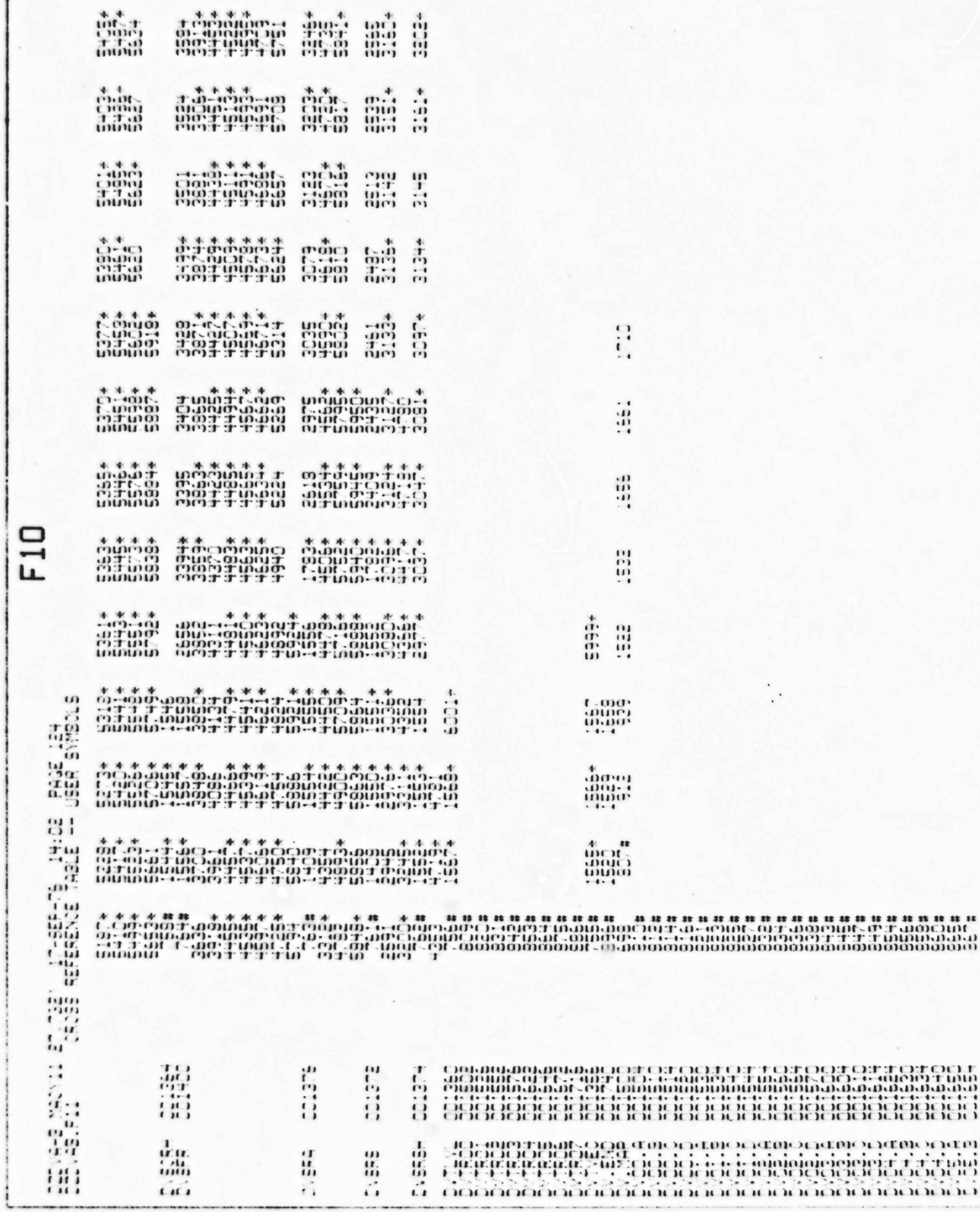
```
6050  033062  006      002                .BYTE   6,2
6051  033064  001264                     $AVR2
6052  033066  006      004                .BYTE   6,4
6053  033070  001272                     $AVR5
6054  033072  006      001                .BYTE   6,1
6055  033074  001270                     $AVR4
6056  033076  000005              DT6:    5
6057  033100  006      003                .BYTE   6,3
6058  033102  001260                     $AVR0
6059  033104  006      001                .BYTE   6,1
6060  033106  001264                     $AVR2
6061  033110  006      004                .BYTE   6,4
6062  033112  001272                     $AVR5
6063  033114  006      001                .BYTE   6,1
6064  033116  001270                     $AVR4
6065  033120  002      001                .BYTE   2,1
6066  033122  001262                     $AVR1
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
```

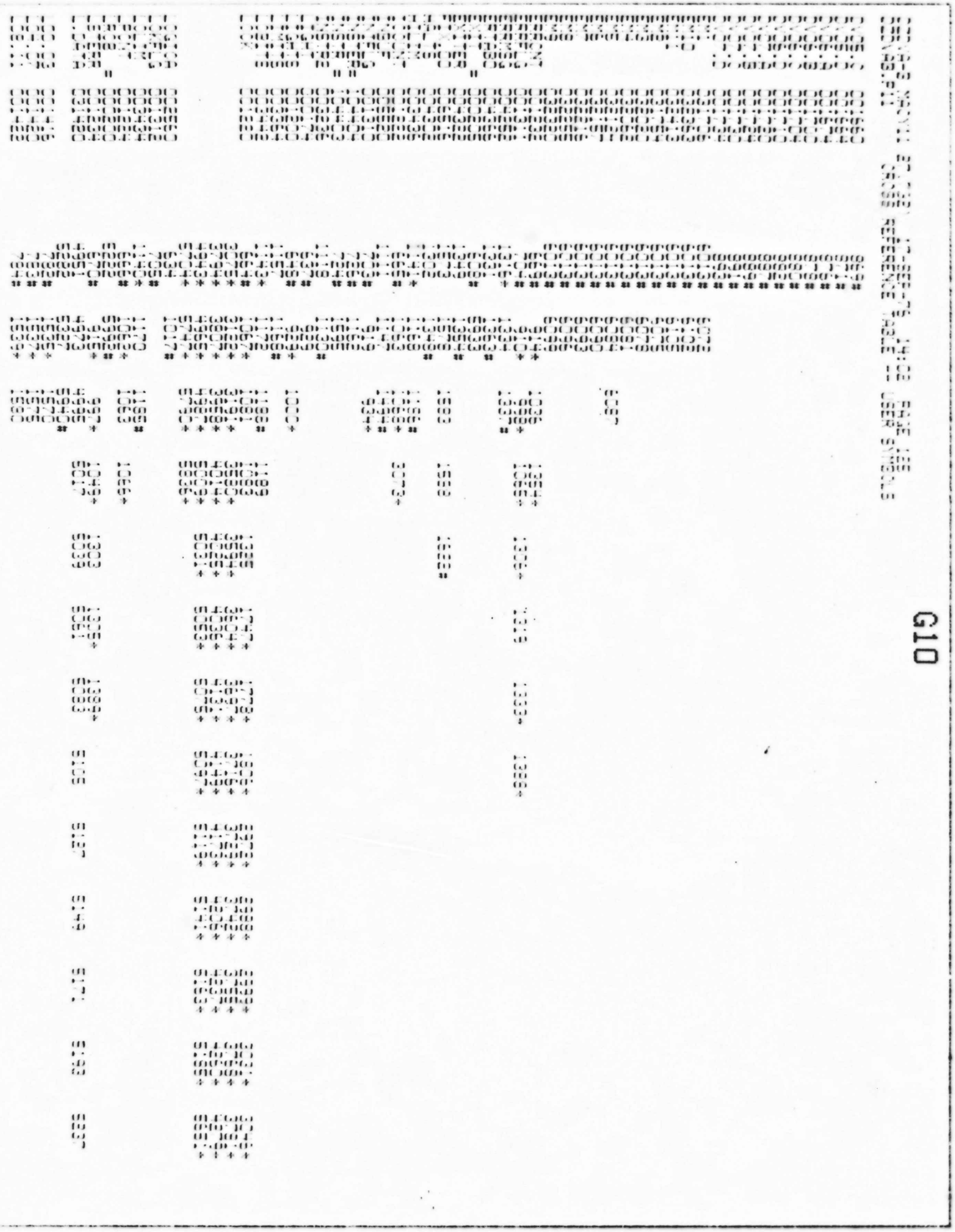
```
EEVVA-B MACY11 27(732)  17-SEP-76  14:08  PAGE 119
EEVVAB.P11      DV11 DEVICE DIAGNOSTICS.   COPYRIGHT 1975  DIGITAL EQUIP. CORP.

 0128  000026  000026                    046     ;HALT  14
 0130  000027  000027                    076
 0132  000042  000042          DIAGDV:
 0134          000001          .END
```

D10



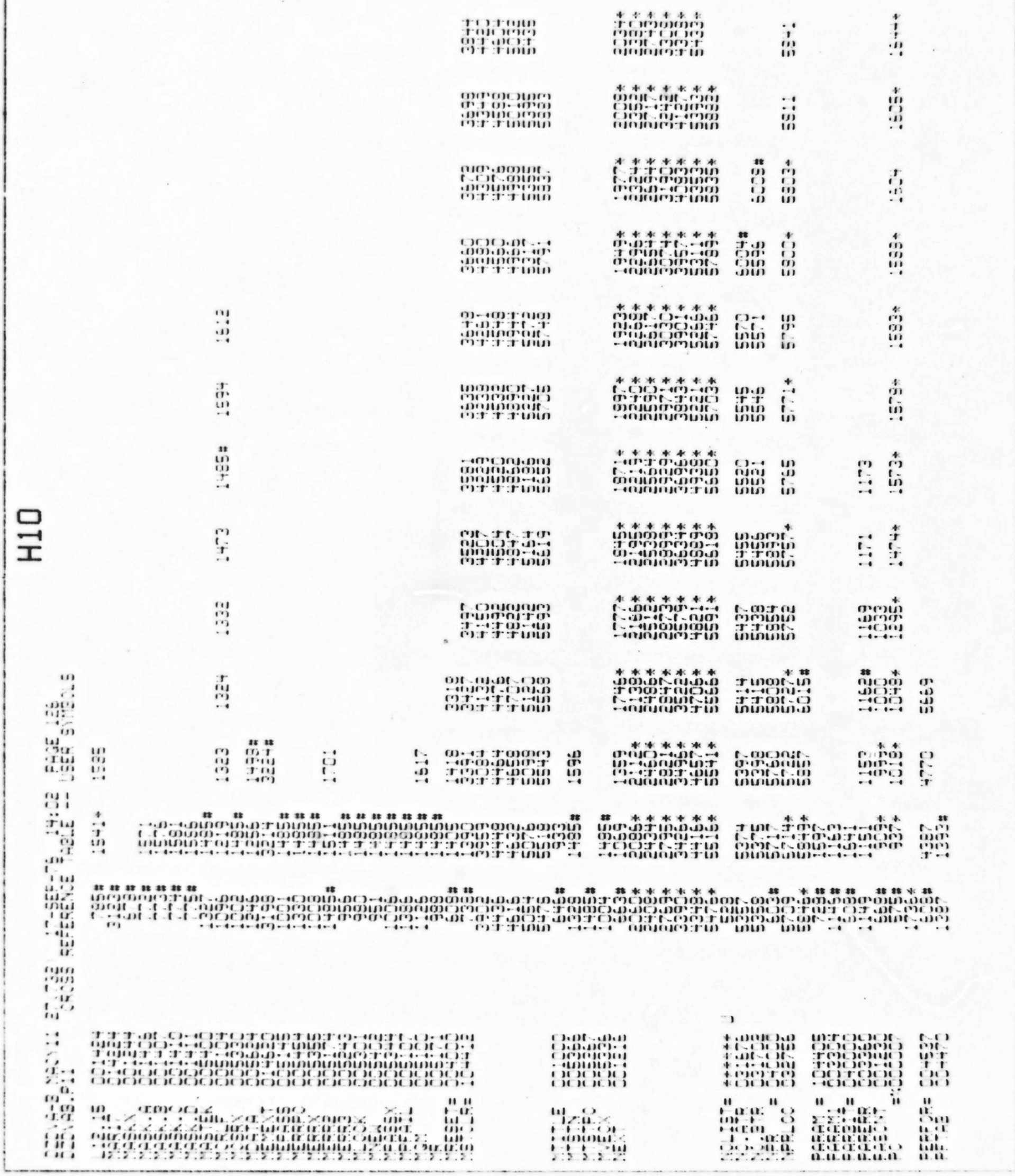
F10

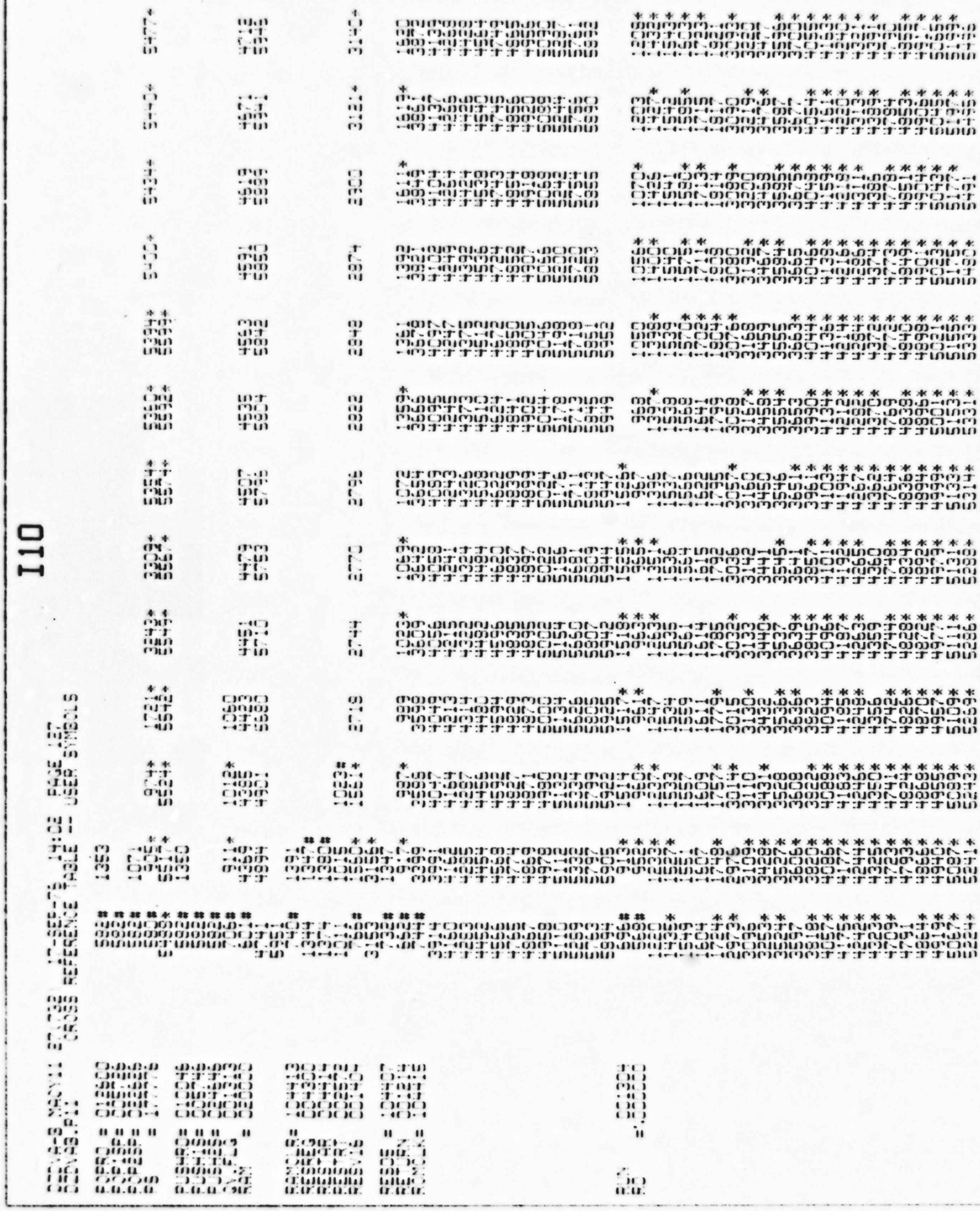




G10

H10





```
                  5501*  5625*  5630*  5654*  5672   5675*  5930*  5932*  5941*  5943*  5946*  5948*  5951*
                  5956   5963*  5965*  5966   5969*  5970*  5971   5974*  5977*  5979*  5981   5987*
R1      =%000001   569#    964*   965    966*   967   1014*  1018   1202   1209*  1221   1225*  1227    1228
                  1229   1230   1262*  1406   1408*  1411*  1414*  1571*  1576*  1593*  1596*  1636*   1631*
                  1636*  1639*  1662*  1664   1666   1669   1671   1684*  1685   1691*  1692   1696*   1712*
                  1713   1714*  1715   1716   1749*  1752   1756*  2974*  2977   3010*  3011   3013*   3033*
                  3037   3041*  3042   3044*  3047   3054*  3055   3077*  3081   3033*  3084   3086*   3094*
                  3097   3103*  3104   3108*  3131*  3134   3145   3161   3163*  3164   3166*  3201*   3202
                  3819*  4045*  4210*  4222   4269   4288   4351*  4357   4363*  4364   4717*  4719    4722
                  4742   4747   4766*  4767   4782*  4786   5225*  5229   5230   5250*  5270*  5274    5275
                  5295*  5315*  5319   5320   5340*  5676*  5923*  5926*  5930   5957   5975*  5976    5977
                  5986*
R2      =%000002   570#   1201   1210*  1572*  1577*  1582*  1587*  1627*  1632*  1637*  1640*  1643*   1650*
                  1657*  1658   1661*  1671*  1672   1673*  1674*  1675*  1676*  1677*  1678*  1679*   1680*
                  1687*  1710*  1722*  1726*  1727*  1728*  1729*  1731*  1732*  1762*  1794*  1796*   1800*
                  1801*  1807*  1818*  1822*  1823*  2975*  2979*  2980   2989*  2990   3000*  3001*   3004*
                  3035*  3038*  3048   3079*  3082*  3098   3123*  3125*  3135*  3137   3141*  3162*   3193*
                  3248*  3250*  3314*  3316*  3432*  3441*  3450*  3459*  3468*  3477*  3501*  3528*   3537*
                  3546*  3555*  3587*  3597*  3608*  3641*  3653*  3665*  3677*  3704*  3708*  3717*   3721*
                  3730*  3734*  3743*  3747*  3756*  3760*  3769*  3773*  3782*  3786*  3795*  3799*   3808*
                  3812*  3819   3820   3822   3824   3829*  3852*  3865*  3876*  3879*  3881*  3905*   3906
                  3911*  3913   3916*  3918   3985*  3986*  3988*  3989*  3990   4007*  4018*  4029*   4040*
                  4045   4046   4055   4063*  4087*  4088   4096   4103   4135*  4138   4163   4177    4187
                  4194   4220   4248   4267   4286   4312*  4313   4325   4352*  4356*  4371*  4374    4660*
                  4666   4673   4691*  4710*  4728   4738*  4744   4753   4754*  4761*  4780*  4786*   4825*
                  4828   4840   4855   4872   4941*  4957   4979   5001   5023   5045   5067   5089    5114
                  5133   5155   5177   5199   5223*  5243   5268*  5288   5313*  5333   5357*  5380*   5379*
                  5680*  5708*  5710*  5712*  5717*  5719*  5751*  5753*  5755*  5760*  5762*  5794*   5796*
                  5798*  5804*  5806*  5808*  5840*  5842*  5844*  5860*  5862*  5854*  5886*  5889*   5891*
                  5893*  5895*  5903*  5906*  5958   5976*  5978*  5979   5985*
R3      =%000003   571#   1105   1112*  1122*  1125*  1127   1131*  1200   1211*  1222   1234*  1235*   1236*
                  1237   1246*  1247*  1252*  1255*  1261*  1622*  1623*  1624   1629   1634   1780*   1781*
                  1786*  1787   1792   1793   1795   1797   1799   1805*  1809   1814   1815   1817    1819
                  1821   1846*  1848*  1849   1853*  1854   1872*  1874*  1875   1879*  1880   1938*   1900*
                  1901   1905*  1906   1924*  1926*  1927   1931*  1932   1950*  1952*  1953   1957*   1958
                  1978*  1980*  1981   1985*  1986   1991*  1992   2009*  2011*  2012   2016*  2017    2035*
                  2037*  2038   2042*  2043   2061*  2063*  2064   2068*  2069   2087*  2089*  2090    2094*
                  2095   2113*  2115*  2116   2120*  2121   2139*  2141*  2142   2146*  2147   2167*   2169*
                  2170   2174*  2175   2180*  2181   2196*  2198*  2199*  2201   2205*  2220*  2223*   2224
                  2241*  2243*  2244   2249*  2250   2269*  2271*  2272   2277*  2278   2297*  2299*   2300
                  2306*  2306   2325*  2327*  2328   2333*  2334   2353*  2355*  2356   2361*  2362    2381*
                  2383*  2384   2389*  2390   2409*  2411*  2412   2416*  2417   2435*  2437*  2438    2442*
                  2443   2461*  2463*  2464   2468*  2469   2487*  2489*  2490   2494*  2495   2513*   2515*
                  2516   2520*  2521   2539*  2541*  2542   2546*  2547   2565*  2567*  2569   2572*   2573
                  2591*  2593*  2594   2598*  2599   2616*  2620*  2621   2626*  2627   2632*  2633    2640*
                  2656*  2658*  2659   2664*  2665   2670*  2671   2678*  2695*  2697   2700*  2701    2718*
                  2720*  2721   2725*  2726   2744*  2746*  2747   2751*  2752   2770*  2772*  2773    2777*
                  2778   2796*  2798*  2799   2803*  2804   2822*  2824*  2825   2829*  2830   2848*   2850*
                  2851   2855*  2856   2874*  2876*  2877   2881*  2882*  2900*  2902*  2903   2907*   2908
                  2930*  2931*  2933*  2934   2939*  2940   2944*  2945*  2946*  2948*  2949   2954*   2955
                  2980*  3005*  2990*  2991   3001*  3002   3048*  3049   3098*  3099   3141*  3148    3204*
                  3205*  3207   3244*  3247*  3249*  3254*  3256   3260   3262   3265   3266   3271    3272
                  3275   3276   3279   3280   3285   3286   3289   3290   3310*  3313*  3315*  3319*   3321
                  3326   3327   3330   3331   3336   3337   3340   3341   3344   3345   3350   3361    3364
                  3365   3429*  3434   3443   3452   3461   3470   3479   3525*  3530   3539   3549    3567
                  3823*  3887*  3989   3995*  3996   3998   4005*  4006*  4016*  4017*  4027*  4029*   4038*
```

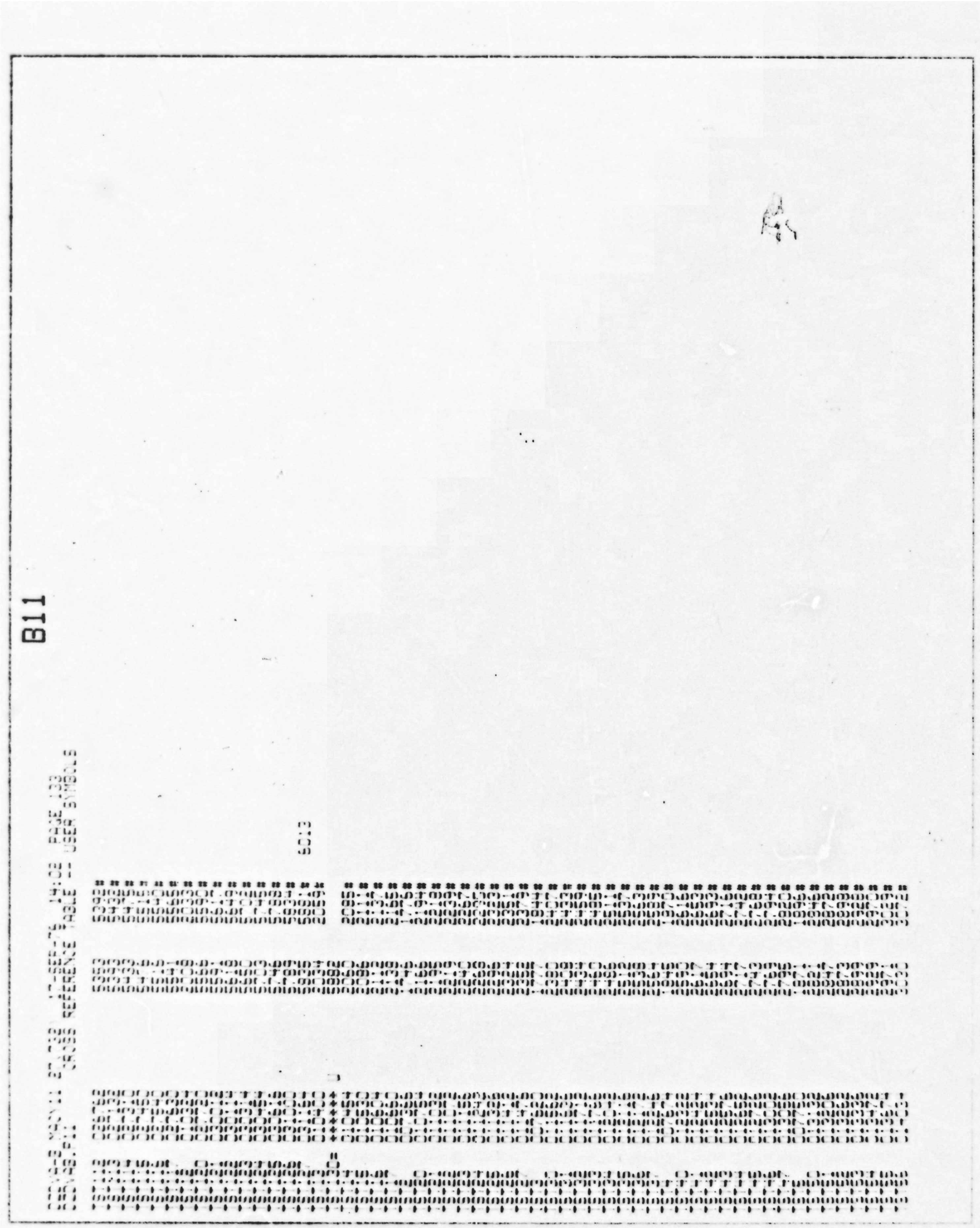
R4 =%000004

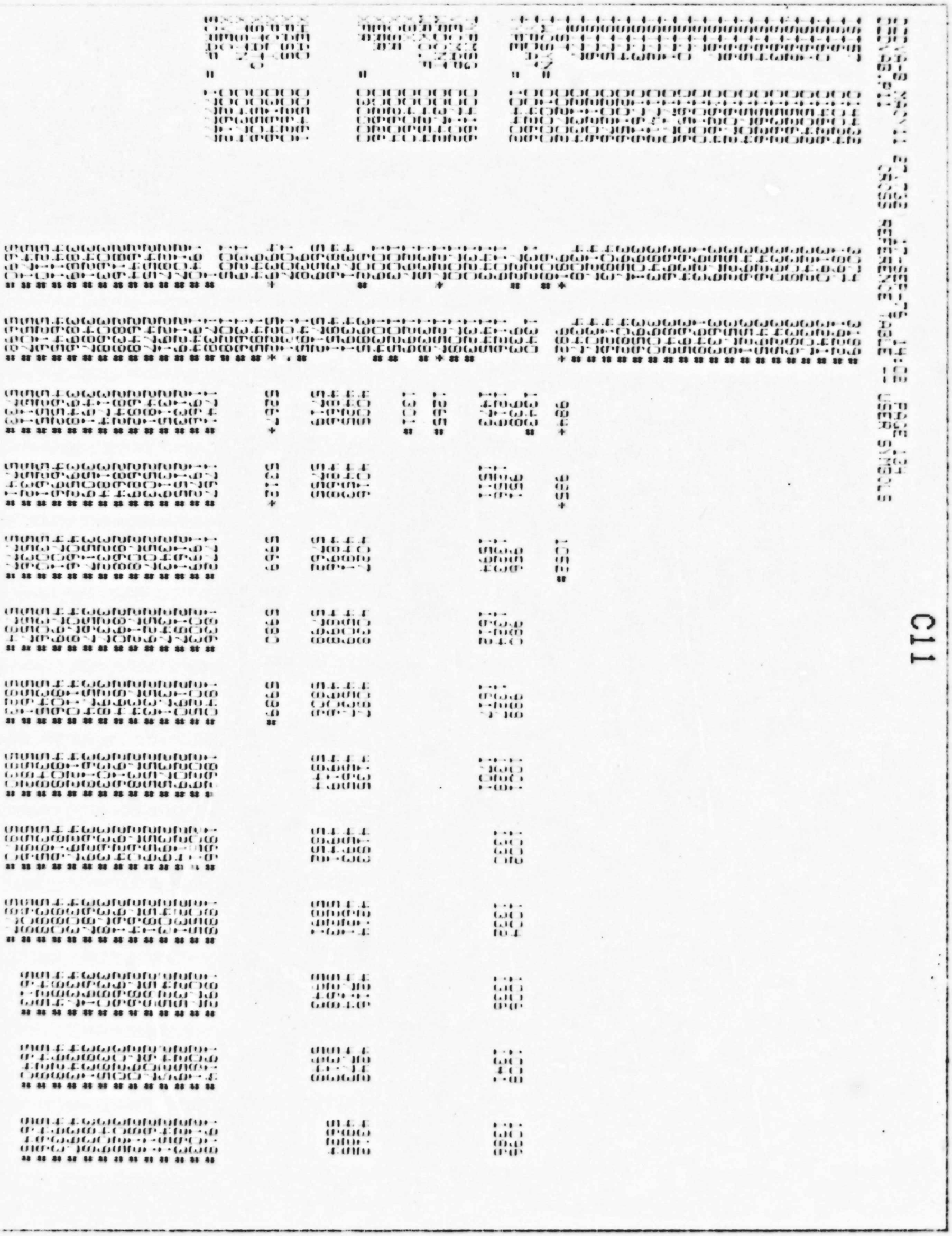
```
 4039*  4048   4057   4211*  4217*  4219*  4221*  4320*  4333*  4353*  4357*  4369*  4373
 4381*  4382   4385*  4659*  4670   4676   4683*  4712*  4721*  4722*  4723   4746*  4747*
 4748   4781*  4794*  4909*  4912*  4951*  4973*  4995*  5017*  5039*  5061*  5083*  5105*
 5120*  5149*  5171*  5193*  5237*  5282*  5327*  5479*  5489*  5656*  5654   5707*  5721
 5450*  5764   5793*  5839*  5885*  5917   5919*  5927*  5929*  5940   5945   5950   5963*
 5722#  1106   1111*  1115*  1116*  1117   1124*  1128   1130*  1138   1147*  1148   1150
 1152   1154*  1155   1156   1177*  1178*  1182*  1199   1212*  1223   1231*  1234   1239*
 1241*  1243*  1260*  1310*  1311*  1312*  1313*  1314*  1315*  1316   1317   1318   1407
 1409*  1410*  1413*  1787*  1788   1809*  1810   1849*  1850   1854*  1856   1875*  1876*
 1880*  1882   1901*  1902   1906*  1908   1927*  1928   1932*  1934   1953*  1954   1958*
 1960   1981*  1986*  1987   1992*  1993   2012*  2013   2017*  2019   2038*  2039   2043*
 2045   2064*  2065   2069*  2071   2090*  2091   2095*  2097   2116*  2117   2121*  2123
 2142*  2143   2147*  2149   2170*  2175*  2176   2181*  2182   2201*  2203   2224*  2285
 2244*  2245*  2246   2250*  2251*  2253   2272*  2273*  2274   2278*  2279*  2281   2300*
 2301*  2302   2306*  2307*  2309   2328*  2329*  2330   2334*  2335*  2337   2356*  2357*
 2358   2362*  2363*  2365   2384*  2385*  2386   2390*  2391*  2393   2412*  2413*  2417*
 2419   2438*  2439   2443*  2445   2464*  2465   2469*  2471   2490*  2491   2495*  2497
 2516*  2517   2521*  2523   2542*  2543   2547*  2549   2568*  2569   2573*  2575*  2594*
 2595   2599*  2601   2621*  2622   2627*  2628   2633*  2634   2659*  2660   2665*  2666
 2671*  2672   2698*  2699*  2700   2701*  2702   2721*  2722   2726*  2728   2747*  2748
 2752*  2754   2773*  2774   2778*  2780   2799*  2800   2804*  2806   2825*  2826   2830*
 2832   2851*  2852   2856*  2858   2877*  2878   2882*  2894   2903*  2904   2908*  2910
 2934*  2935   2940*  2941   2949*  2950   2955*  2956   2978*  2979   2981   2985*  2987*
 2989   2991   2996*  2998*  3000   3002   3007*  3034*  3038   3039*  3045*  3049   3053*
 3078*  3082   3087*  3095*  3099   3106*  3132*  3135   3148   3158*  3162   3194*  3199
 3207   3216*  3256*  3257   3262*  3266*  3267*  3268   3272*  3276*  3280*  3282   3290*
 3292   3321*  3322   3327*  3331*  3332*  3333   3337*  3341*  3345*  3347   3355*  3357
 3434*  3435*  3437   3443*  3444*  3446   3452*  3453*  3455   3461*  3462*  3464   3470*
 3471*  3473   3479*  3480*  3482   3503*  3504   3530*  3531*  3533   3539*  3540*  3542
 3548*  3549*  3551   3557*  3558*  3560   3589*  3590   3599*  3600   3610*  3611   3643*
 3644   3655*  3656   3667*  3668   3679*  3680   3825*  3826   3853*  3854   3856*  3868
 3882*  3884   3907*  3908   3915*  3922*  3924*  3925   3935*  3936   3941*  3942   3992*
 4050*  4051   4060*  4091*  4098*  4105*  4140*  4142   4152*  4166*  4179*  4180   4189*
 4196*  4197   4222*  4223   4230*  4250*  4252   4269*  4271   4288*  4290   4316*  4327*
 4328   4355*  4358   4359   4376*  4377   4430*  4431*  4433   4439*  4440*  4442   4458*
 4459*  4461   4467*  4468*  4470   4486*  4487*  4489   4495*  4496*  4498   4514*  4515*
 4517   4523*  4524*  4526   4542*  4543*  4545   4551*  4552*  4554   4570*  4571*  4573
 4579*  4580*  4582   4598*  4599*  4601   4607*  4608*  4610   4626*  4627*  4629   4635*
 4636*  4638   4663*  4665*  4666   4674*  4675*  4679   4687*  4689*  4689   4714*  4727
 4730*  4731*  4732   4739*  4752   4756*  4757*  4758   4762*  4764   4786*  4787   4931*
 4843*  4844   4857*  4858   4875*  4908*  4911*  4914   4960*  4961   4982*  4983   5004*
 5005   5026*  5027   5048*  5049   5070*  5071   5092*  5093   5114*  5115   5136*  5137
 5158*  5159   5180*  5181   5202*  5203   5246*  5247   5291*  5292   5336*  5337   5664*
 5665   5721*  5723   5764*  5767   5811*  5912   5857*  5958*  5859   5996*  5997   5900
 5907*
 5908
 5910

R5  =%000005

 573#  1089   1090*  1094   1099   1101*  1137   1139*  1140   1141   1142   1143   1144
 1145   1146*  1155*  1158*  1159*  1160*  1168   1170   1172   1178   1179*  1183*  1188
 1213*  1224   1232*  1244*  1259*  1308*  1309*  1310   1312   1795*  1786   1798   1808*
 1810   1847*  1848   1850   1853   1855*  1856   1873*  1874   1876   1879   1981*  1982
 1899*  1900   1902   1905   1907*  1908   1925*  1926   1928   1931   1933*  1934   1951*
 1952   1954   1957   1959*  1960   1979*  1984*  1985   1987   1990*  1993   2010*  2011
 2013   2016   2018*  2019   2036*  2037   2039   2042   2044*  2045   2062*  2063   2065
 2068   2070*  2071   2088*  2089   2091   2094   2096*  2097   2114*  2115   2117   2120
 2122*  2123   2140*  2141   2143   2146   2148*  2149   2168*  2173*  2174   2176   2179*
 2182   2197*  2202   2222*  2225   2242*  2243   2246   2249   2252*  2253   2270*  2271
 2274   2277   2280*  2281   2298*  2299   2302   2305   2308*  2309   2326*  2327   2330
```

```
SETVEC  031654          5356    5366    5373    5396    5406    5413    5436    5447    5454    5481    5519    5544    5569
                        5594    5997#
SIMBCC  031474          4947    4969    4991    5013    5035    5057    5079    5101    5123    5145    5167    5189    5233
                        5278    5323    5956#
SKIP   = 000000         1828    1832    6013#
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    1308    1351    1360*   1382*   1396*
                        1400    1406*   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*   4728*   4738    4739    4740    4751*   4752*   4753*   4761    4762    4763    4769*   4772
                        5379*   5419*   5460*   5528*   5553*   5578*   5604*   5917*   5929    5956*   5957*   5958*   5985
                        5986    5987
SPACNT= 003745          1229*   1250    1253*   1267#
SSWR    000176           639#    930
STACK  = 001200          581#    905     975     1360    1382    5379    5419    5460
STAT    001236           575#
SVOS    003452          1198#
SWR     001202           650#    925     930*    935     949     954     959     978    1042    1051    1058    1079    1092
                        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  001506           811#
SYNA01  001532           822#
SYNA02  001556           833#
SYNA03  001602           844#
SYNA04  001626           855#
SYNA05  001652           866#
SYNA06  001676           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           676#
```

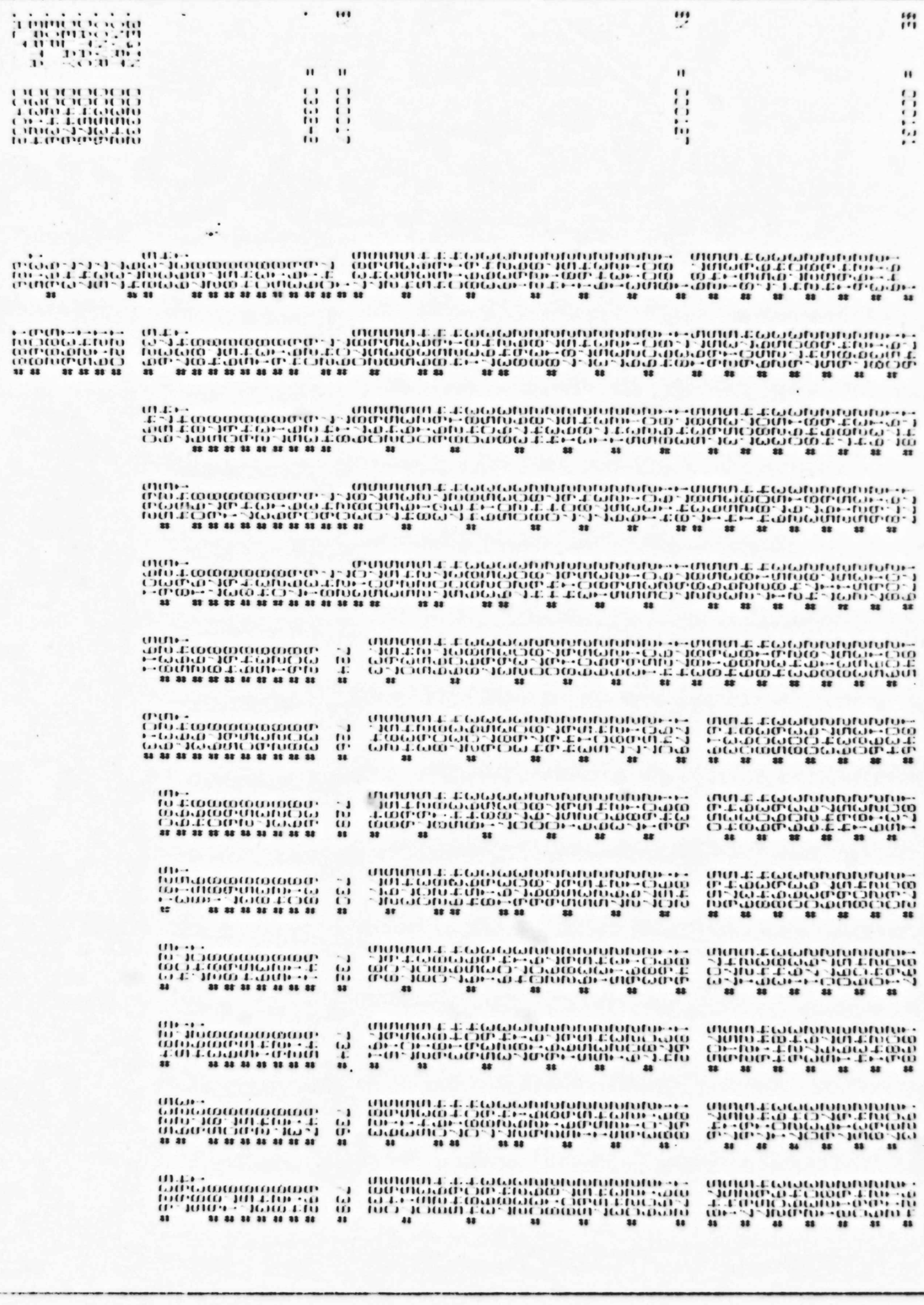
```
SYNC00  001516           815#
SYNC01  001542           826#
SYNC02  001566           837#
SYNC03  001612           848#
SYNC04  001636           859#
SYNC05  001662           870#
SYNC06  001706           881#
SYNC07  001732           892#
SYNC2A  001426           787#   1536*
SYNC2B  001430           788#   1538*
SYNC2C  001432           789#   1540*
SYNC2D  001434           790#   1542*
SYND00  001522           817#
SYND01  001546           828#
SYND02  001572           839#
SYND03  001616           850#
SYND04  001642           861#
SYND05  001666           872#
SYND06  001712           883#
SYND07  001736           894#
S.C   = 050000           604#   3393   3585   3596   3606   3639   3651   3663   3675   3818   3848   3859   3871
                        3911    4044   4054   4265   4284   4321   4323   5903
TEMP    005562          1233    1383*  1384*  1496#
TEMP1   001246           679#    939*   940    945*  1487   1718*  1719*  3196*  3217*  4779*  4785   4794*  4795
                        5658*   5663   5671*  5674*  5678*  5959*  5982*
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*  1922*
                        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*  2898*  2928*  2970*  3029*
                        3072*   3189*  3241*  3307*  3389*  3425*  3495*  3521*  3578*  3633*  3695*  3842*  3900*
                        3956*   4082*  4126*  4307*  4347*  4415*  4656*  4705*  4820*  4888*  4937*  5220*  5265*
                        5310*   5352*  5392*  5432*  5475*  5515*  5540*  5565*  5590*  5617*  5649*  5702*  5745*
                        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#
```

B11



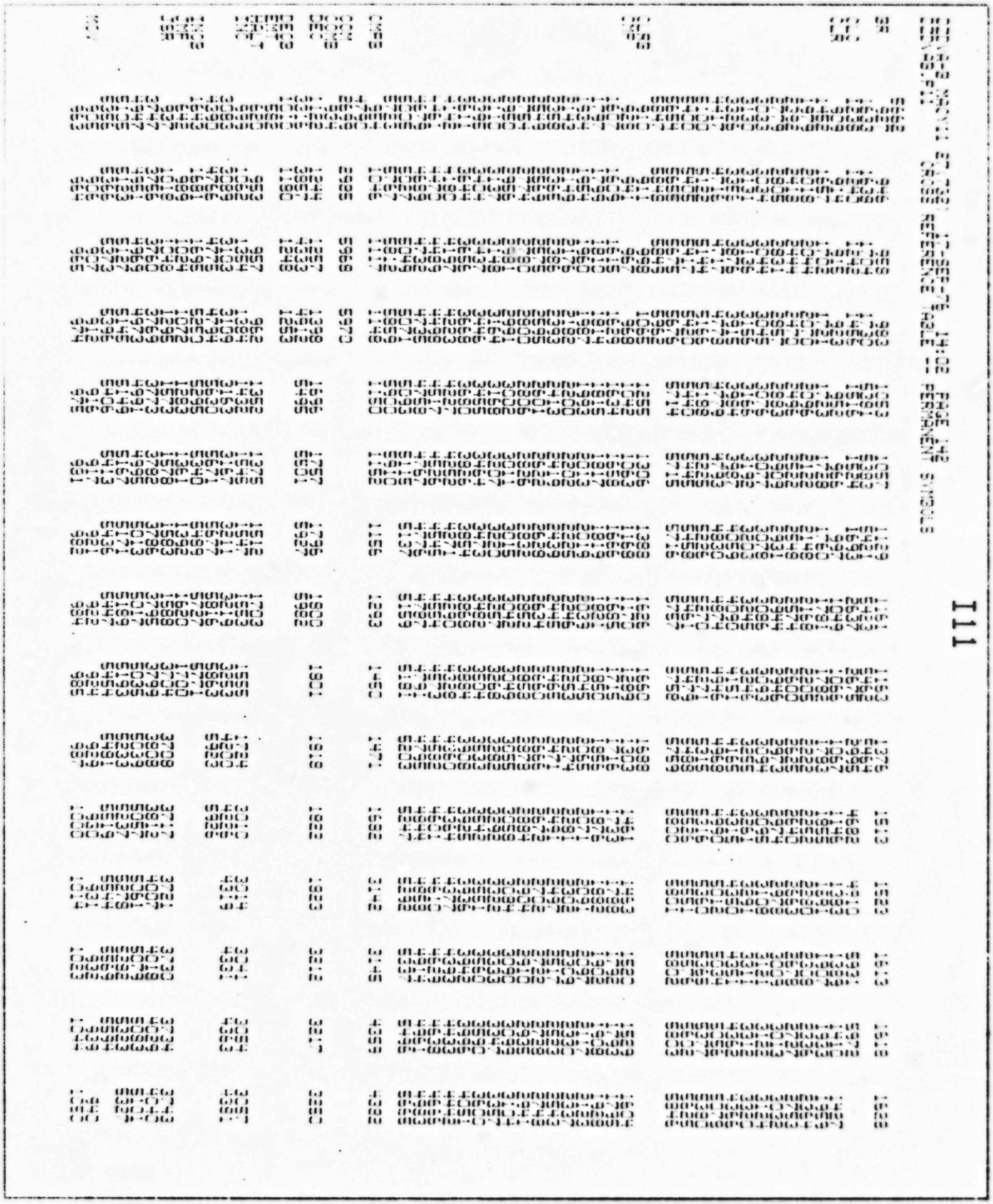


C11



D11

F11



```
                4371  4372  4374  4376  4385  4386  4389  4391  4415  4416  4421  4422  4423  4425  4427
                4429  4430  4432  4437  4438  4439  4441  4449  4450  4451  4453  4455  4457  4458  4460
                4465  4466  4467  4469  4477  4478  4479  4481  4483  4485  4486  4488  4489  4490  4492
                4497  4505  4506  4507  4509  4511  4513  4514  4516  4521  4522  4523  4525  4526  4527
                4536  4570  4572  4577  4578  4579  4581  4589  4591  4593  4594  4595  4596  4597  4598
                4605  4606  4607  4609  4610  4618  4619  4621  4623  4625  4626  4627  4628  4629  4630
                4637  4687  4688  4689  4690  4691  4692  4693  4694  4695  4696  4697  4698  4699  4700
                4709  4714  4730  4735  4738  4740  4741  4744  4745  4750  4752  4753  4754  4755  4756
                4763  4764  4765  4766  4767  4769  4772  4773  4780  4782  4783  4784  4785  4786  4787
                4826  4828  4829  4830  4831  4834  4836  4838  4840  4842  4843  4844  4845  4846  4847
                4863  4864  4865  4866  4867  4868  4870  4872  4875  4877  4879  4880  4881  4882  4883
                4900  4901  4902  4903  4904  4905  4906  4908  4909  4911  4912  4913  4914  4915  4916
                4945  4955  4956  4957  4958  4959  4960  4965  4967  4968  4969  4970  4971  4972  4973
                5003  5004  5005  5006  5007  5011  5012  5023  5025  5026  5027  5028  5029  5030  5031
                5055  5065  5066  5067  5068  5069  5070  5075  5076  5077  5078  5079  5080  5081  5082
                5113  5114  5115  5116  5117  5118  5119  5131  5133  5134  5135  5136  5137  5138  5139
                5166  5175  5176  5177  5178  5179  5180  5185  5186  5187  5188  5189  5190  5191  5192
                5243  5273  5274  5275  5276  5230  5231  5232  5233  5234  5235  5236  5237  5238  5239
                5319  5320  5321  5322  5323  5324  5325  5326  5327  5328  5329  5330  5331  5332  5333
                5393  5394  5395  5396  5401  5403  5404  5405  5406  5407  5408  5409  5410  5411  5412
                5480  5485  5486  5487  5488  5489  5490  5491  5492  5493  5494  5495  5496  5497  5498
                5566  5567  5568  5569  5570  5516  5573  5574  5575  5576  5577  5578  5579  5580  5581
                5649  5650  5651  5652  5653  5654  5655  5656  5657  5658  5659  5660  5661  5662  5663
                5707  5708  5709  5710  5711  5712  5713  5714  5715  5716  5717  5718  5719  5720  5721
                5750  5753  5754  5755  5756  5757  5758  5759  5760  5761  5762  5763  5764  5765  5766
                5798  5804  5806  5808  5810  5811  5812  5813  5814  5815  5816  5817  5818  5819  5820
                5850  5852  5854  5856  5857  5881  5882  5883  5884  5885  5886  5887  5888  5889  5890
                5896  5897  5903  5906  5907  5908  5917  5919  5921  5924  5929  5930  5931  5932  5933
                5945  5946  5948  5950  5951  5956  5957  5958  5959  5960  5961  5963  5964  5965  5966
                5976  5984  5985  5986  5987  5988  5999  6000  6001
MOVB             908   913   959   960  1012  1013  1099  1115  1143  1144  1228  1229  1238  1237  1247
                1288  1300  1333  1627  1631  1632  1636  1637  1639  1640  1691  1692  1699  1719  3133
                2938  2954  2976  2977  3036  3037  3047  3080  3081  3096  3097  3133  3134  3160
                3161  3200  3201  3202  4356  4357  4718  4719  4784  4785  6000  6001
NOP              926   981   982  1019  1020  1021  1022  1753  1754  3400  3852  3873  5362  5364  5371
                5402  5404  5411  5442  5444  5452  5525  5550  5575  5600
RESET            996  1016  1655  3252
ROL             1457  1458  1459  2638  2676  2996  3007  3158  4683  4911
ROLB            1517  1526  1694
RORB            1239  1241  1243  1719  5932  5964  5973
RORB            1430
RTI             1084  1102  1132  1184  1204  1214  1264  1362  1401  1415  1419  1423  1436  1706  1735
                1764  5500  5504  5529  5554  5579  5605  6006  6010
RTS             1474  1644  1736  3829  4063  5953  5988  6002
SEC             3006  3157
SUB             1277  1309  3216
SWAB            5766  5850
TRAP            7180   720   722   724   726   728   730   732   734   736   738   740   742   744   746
TST              935   947   976  1081  1321  1325  1334  1339  1343  1348  1592  1614  1624  1664  1663
                1672  1752  3153  3260  3265  3271  3275  3279  3285  3286  3289  3325  3330  3336  3340
                3344  3350  3351  3354  4676  5491  5620  5627  5633  5634  5661
TSTB             919   935  1054  1060  1094  1097  1113  1119  1250  1298  1319  1446  1453  1467  1503
                1683
```

```
.ASCIZ    646  1476  1483  1485  6013
.BLKB     693   693   694   696   696
.BLKW     631   632   633   808   809   810   811   812   813   814   815   816   817   819   820
          824   825   826   827   828   829   830   831   832   833   834   835   836   838   839
          839   840   841   842   842   843   844   845   846   847   848   849   850   852   853
          854   855   856   857   858   859   860   861   863   864   865   866   867   869   870
          871   872   873   874   875   876   877   878   879   880   881   882   883   885   886
          887   888   889   890   891   892   893   894  1437  5235  5236  5238  5239  5240  5241
         5243  5244  5245  5246  5247  5248  5249  5250  5251  5252  5253  5254  5255  5256  5257
.BYTE     703   704   705   706   772   773   774   775   777   778   779   780  1026  1029  1032
         1035  1364  1367  1394  1480  1486  1488  1601  1602  3224  3225  3226  3227  3228  3360
         5360  5376  5399  5409  5416  5439  5450  5457  5484  5520  5547  5572  5597  6017  6019
         6022  6024  6026  6028  6030  6033  6035  6037  6040  6042  6044  6046  6050  6052  6054
         6057  6059  6061  6063  6065
.ENABL    542
.END      610
.ENDC    1738  1739  1742  1747  1768  1773  1778  1837  1841  1846  1863  1867  1872  1889  1893
         1891  1914  1919  1924  1941  1945  1950  1967  1973  1978  2000  2004  2009  2026  2030
         2034  2052  2056  2061  2078  2082  2087  2104  2108  2113  2130  2134  2133  2156  2160
         2164  2188  2191  2196  2210  2215  2220  2224  2232  2236  2241  2260  2264  2259  2304
         2308  2297  2316  2320  2325  2344  2348  2353  2372  2376  2381  2400  2404  2409  2434
         2438  2436  2452  2456  2461  2478  2482  2487  2504  2508  2513  2530  2534  2539  2564
         2568  2565  2582  2586  2591  2608  2612  2618  2645  2649  2655  2683  2690  2695  2712
         2716  2713  2719  2735  2739  2744  2761  2765  2770  2791  2796  2813  2817  2817  2842
         2846  2843  2848  2865  2869  2874  2891  2895  2900  2917  2925  2930  2938  2967  2972
         3026  3026  3031  3064  3069  3075  3175  3185  3191  3233  3238  3243  3254  3299  3304
         3314  3319  3364  3379  3381  3386  3391  3410  3422  3427  3434  3443  3452  3461  3470
         3484  3488  3492  3497  3510  3518  3523  3530  3539  3548  3557  3568  3575  3600  3619
         3630  3635  3688  3692  3697  3832  3839  3844  3890  3897  3902  3949  3953  3958  4067
         4079  4084  4111  4123  4128  4298  4304  4309  4338  4344  4349  4398  4412  4417  4646
         4653  4658  4697  4702  4707  4800  4807  4809  4817  4822  4881  4885  4890  4929  4934
         4939  5211  5217  5222  5256  5262  5267  5301  5307  5312  5346  5349  5354  5388  5389
         5394  5426  5429  5434  5467  5472  5477  5508  5512  5517  5533  5537  5542  5562  5563
         5567  5583  5587  5592  5609  5614  5619  5641  5646  5651  5691  5699  5704  5723  5724
         5729  5734  5742  5747  5765  5767  5773  5778  5785  5790  5811  5812  5816  5819  5834
         5831  5836  5857  5859  5863  5866  5871  5878  5883  6109
.EQUIV    688
.EVEN     697   708  1478  1484  1485  1490  3229  6013
.IF       688  1738  1741  1746  1767  1772  1777  1836  1840  1845  1862  1866  1871  1888  1892
         1897  1914  1918  1923  1940  1944  1949  1966  1972  1977  1999  2003  2008  2025  2029
         2034  2051  2055  2060  2077  2081  2086  2103  2107  2112  2129  2133  2138  2156  2161
         2166  2187  2190  2195  2209  2214  2219  2222  2231  2235  2240  2259  2263  2268  2287
         2291  2296  2315  2319  2324  2343  2347  2352  2371  2375  2380  2399  2403  2408  2426
         2430  2434  2451  2455  2460  2477  2481  2486  2503  2507  2512  2529  2533  2538  2555
         2559  2564  2581  2585  2590  2607  2611  2617  2644  2648  2654  2682  2689  2694  2697
         2708  2712  2717  2734  2738  2743  2760  2764  2769  2786  2790  2795  2812  2816  2821
         2838  2842  2847  2864  2868  2873  2890  2894  2899  2916  2924  2929  2961  2966  2971
         3020  3025  3030  3063  3068  3074  3174  3184  3190  3232  3237  3242  3252  3299  3303
         3308  3313  3363  3378  3380  3385  3390  3409  3421  3426  3433  3442  3451  3460  3469
         3478  3487  3491  3496  3509  3517  3522  3529  3538  3547  3556  3567  3574  3579  3618
         3629  3634  3687  3691  3696  3831  3838  3843  3889  3896  3901  3948  3952  3957  4066
         4078  4083  4110  4122  4127  4297  4303  4308  4337  4343  4348  4397  4411  4416  4645
         4652  4657  4696  4701  4706  4799  4806  4808  4816  4821  4880  4884  4889  4928  4933
         4938  5210  5216  5221  5255  5261  5266  5300  5306  5311  5345  5348  5353  5385  5389
         5393  5425  5428  5433  5466  5471  5476  5507  5511  5516  5532  5536  5541  5557  5562
         5566  5582  5586  5591  5608  5613  5618  5640  5645  5650  5690  5698  5703  5717  5733
```

```
                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
                1872    1889    1892    1897    1898    1915    1918    1923    1924    1941    1944    1949    1950    1967    1972
                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    2167
                2188    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    2460
                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    2718    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    3068    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    3488    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    3958    4067
                4078    4083    4084    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    5388    5393
                5394    5426    5428    5433    5434    5467    5471    5476    5477    5508    5511    5516    5517    5533    5536
                5541    5542    5558    5561    5566    5567    5583    5586    5591    5592    5609    5613    5618    5619    5641
                5646    5650    5651    5691    5698    5703    5704    5722    5723    5734    5741    5746    5747    5765    5767
                5778    5784    5789    5790    5803    5812    5824    5830    5835    5836    5949    5857    5871    5877    5882
.IFT             528    1738    1741    1767    1772    1836    1840    1862    1866    1888    1892    1914    1918    1940    1944
                1966    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
                2760    2764    2786    2790    2812    2816    2838    2842    2864    2868    2890    2894    2916    2924    2961
                2966    3020    3025    3063    3068    3174    3184    3232    3237    3252    3298    3303    3318    3363    3378
                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    4652    4696    4701    4799
                4806    4808    4816    4880    4884    4928    4933    5210    5216    5255    5261    5300    5306    5345    5348
                5385    5388    5425    5428    5466    5471    5507    5511    5532    5536    5557    5561    5582    5586    5608
                5613    5640    5645    5690    5698    5717    5723    5733    5741    5760    5765    5777    5784    5803    5812
                5823    5830    5849    5857    5870    5877
.IFTF            528
.IIF            1746    1747    1777    1778    1845    1846    1848    1850    1855    1871    1872    1874    1876    1881    1897
                1898    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    2268    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    2857    2873    2874    2876    2878    2883    2899    2900
                2902    2904    2909    2929    2930    2971    2972    3030    3031    3073    3075    3190    3191    3242    3243
                3308    3309    3390    3391    3426    3427    3496    3497    3522    3523    3579    3580    3589    3590    3593
```

```
                3600    3610    3611    3634    3635    3643    3655    3667    3679    3695    3697    3843    3844    3901    3902
                3957    3958    4083    4084    4127    4128    4308    4309    4348    4349    4416    4417    4657    4658    4706
                4787    4821    4822    4889    4890    4938    4939    5221    5222    5266    5267    5311    5312    5353    5354
                5366    5393    5394    5406    5433    5434    5446    5476    5477    5516    5517    5526    5541    5542    5551
                5566    5567    5576    5591    5592    5601    5618    5619    5650    5651    5695    5696    5697    5698    5703
                5704    5709    5710    5712    5713    5722    5723    5727    5728    5738    5739    5740    5746    5747    5753
                5755    5756    5771    5782    5783    5784    5789    5790    5795    5796    5798    5811    5828    5829    5835
                5836    5841    5844    5882    5883
.IRP             718     720     722     724     726     728     730     732     734     736     738     740     742     744     746
                 808    1738    1743    1746    1767    1774    1777    1835    1836    1842    1845    1862    1868    1871    1888
                1894    1897    1914    1920    1923    1940    1946    1949    1966    1974    1977    1998    1999    2005    2008
                2025    2031    2034    2051    2057    2060    2077    2083    2086    2103    2109    2112    2129    2135    2138
                2155    2163    2166    2187    2192    2195    2209    2216    2219    2230    2231    2237    2240    2259    2266
                2268    2287    2293    2296    2315    2321    2324    2343    2349    2352    2371    2377    2380    2398    2399
                2405    2408    2425    2431    2434    2451    2457    2460    2477    2483    2486    2503    2509    2512    2513
                2535    2538    2555    2561    2564    2581    2587    2590    2607    2614    2617    2644    2651    2654    2682
                2691    2694    2707    2708    2714    2717    2734    2740    2743    2760    2766    2769    2786    2792    2795
                2812    2818    2821    2838    2844    2847    2864    2870    2873    2890    2896    2899    2916    2926    2929
                2961    2968    2971    3020    3027    3030    3063    3070    3074    3174    3187    3190    3232    3239    3242
                3298    3305    3308    3363    3380    3387    3390    3409    3423    3426    3487    3493    3496    3509    3519
                3522    3567    3576    3579    3618    3631    3634    3687    3693    3696    3831    3840    3843    3889    3898
                3901    3948    3954    3957    4066    4080    4083    4110    4124    4127    4297    4305    4308    4337    4345
                4348    4397    4413    4416    4645    4654    4657    4696    4703    4706    4799    4808    4818    4821    4880
                4886    4889    4928    4935    4938    5210    5218    5221    5255    5263    5266    5300    5308    5311    5345
                5350    5353    5385    5390    5393    5425    5430    5433    5466    5473    5476    5507    5513    5516    5532
                5538    5541    5557    5563    5566    5582    5588    5591    5608    5615    5618    5640    5647    5650    5690
                5700    5703    5733    5743    5746    5777    5786    5789    5823    5832    5835    5870    5879    5882    6013
.LIST               1     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     897     990    1038    1485
                1738    1741    1748    1767    1772    1779    1836    1840    1846    1862    1866    1872    1888    1892    1898
                1914    1918    1924    1940    1944    1950    1966    1972    1978    1999    2003    2009    2025    2029    2035
                2051    2055    2061    2077    2081    2087    2103    2107    2113    2129    2133    2139    2155    2161    2167
                2187    2190    2196    2209    2214    2220    2231    2235    2241    2259    2263    2269    2287    2291    2297
                2315    2319    2325    2343    2347    2353    2371    2375    2381    2399    2403    2409    2425    2429    2435
                2451    2455    2461    2477    2481    2487    2503    2507    2513    2529    2533    2539    2555    2559    2565
                2581    2585    2591    2607    2611    2618    2644    2648    2655    2682    2689    2695    2708    2712    2719
                2734    2738    2744    2760    2764    2770    2786    2790    2796    2812    2816    2822    2838    2842    2848
                2864    2868    2874    2890    2894    2900    2916    2924    2930    2961    2966    2973    3020    3025    3032
                3063    3068    3076    3174    3184    3192    3232    3237    3243    3298    3303    3309    3363    3378    3380
                3385    3391    3409    3421    3427    3487    3491    3497    3509    3517    3523    3567    3574    3581    3618
                3629    3635    3687    3691    3698    3831    3838    3844    3889    3896    3902    3948    3952    3959    4066
                4078    4084    4110    4122    4128    4297    4303    4309    4337    4343    4349    4397    4411    4417    4645
                4652    4658    4696    4701    4707    4799    4806    4808    4816    4822    4880    4884    4890    4928    4933
                4939    5210    5216    5222    5255    5261    5267    5300    5306    5312    5345    5348    5354    5385    5388
                5394    5425    5428    5434    5466    5471    5477    5507    5511    5517    5532    5536    5542    5557    5561
                5567    5582    5586    5592    5608    5613    5619    5640    5645    5652    5690    5698    5705    5733    5741
                5748    5777    5784    5791    5823    5830    5837    5870    5877    5883    6013
.MACRO              1     528
.NLIST              1     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     897     990    1038    1485
                1738    1741    1748    1767    1772    1779    1836    1840    1846    1862    1866    1872    1888    1892    1898
                1914    1918    1924    1940    1944    1950    1966    1972    1978    1999    2003    2009    2025    2029    2035
                2051    2055    2061    2077    2081    2087    2103    2107    2113    2129    2133    2139    2155    2161    2167
                2187    2190    2196    2209    2214    2220    2231    2235    2241    2259    2263    2269    2287    2291    2297
                2315    2319    2325    2343    2347    2353    2371    2375    2381    2399    2403    2409    2425    2429    2435
                2451    2455    2461    2477    2481    2487    2503    2507    2513    2529    2533    2539    2555    2559    2565
```

B12

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