DH11

DH11 DIAGNOSTIC

MD-11-DZDHM-A

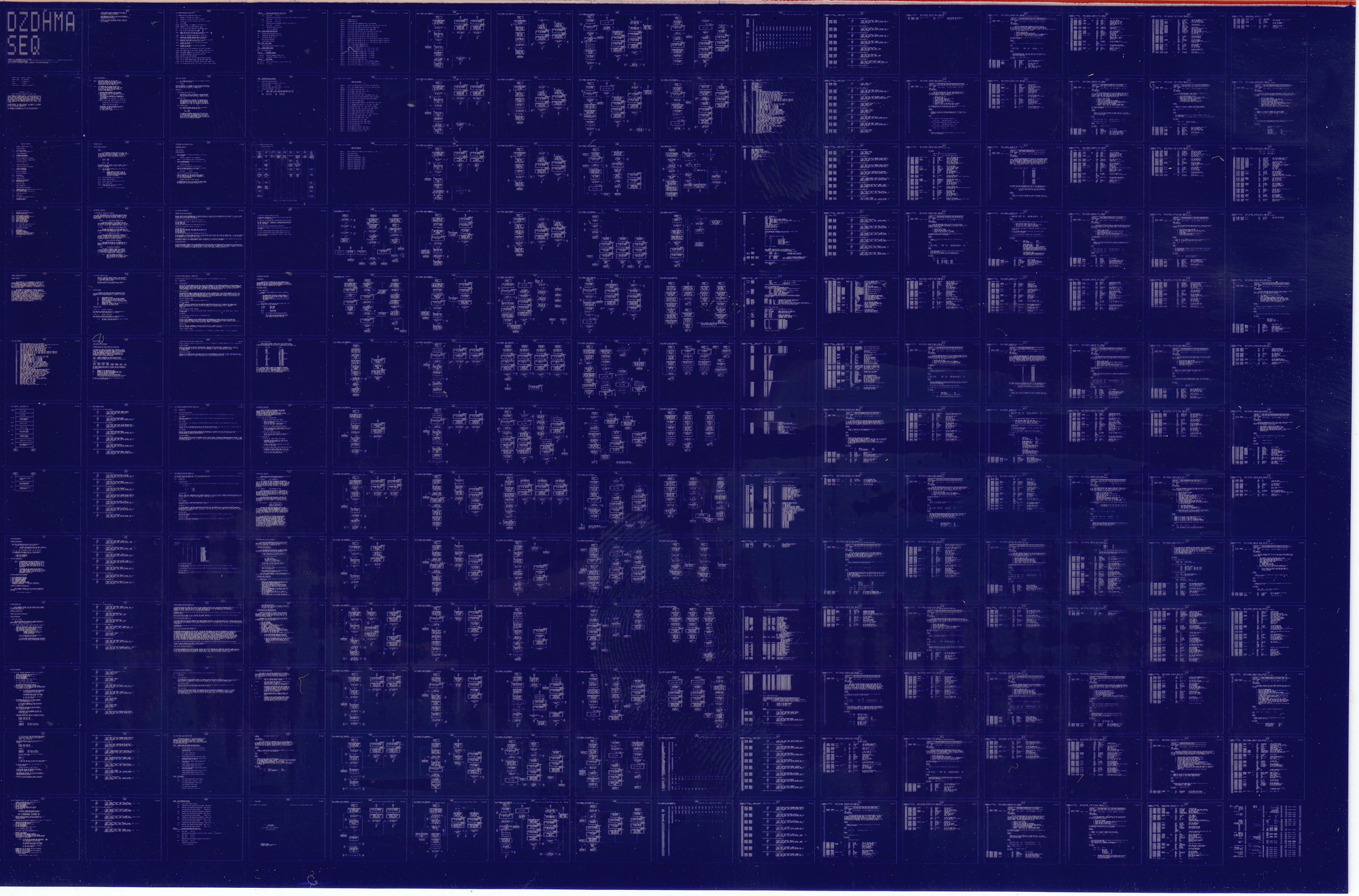
EP-DZDHM-A-DL-A

FEB 1976

FICHE 1 OF 2

digital

Made In U.S.A.



DH11

DH11 DIAGNOSTIC

MD-11-DZDHM-A

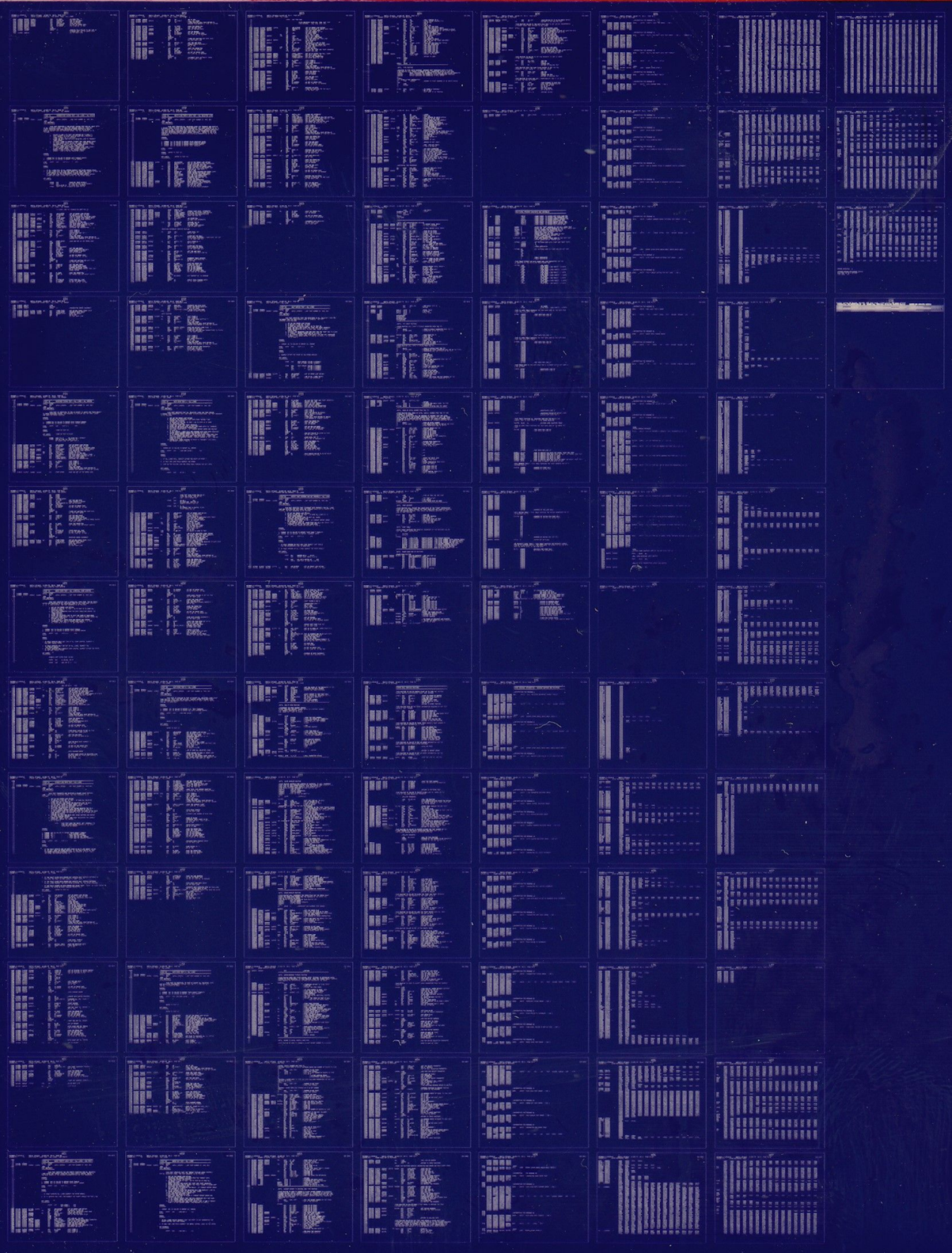
EP-DZDHM-A-DL-A

FEB 1976

digital

Made In U.S.A.

FICHE 2 OF 2



```
 4040  012314  004767  012014          JSR     PC,SAPS         ;GO SAVE PSW
 4041  012320  012704  037400          MOV     #37400,R4       ;SET UP S/B DATA
 4042  012324  062702  000016          ADD     #SSR,R2         ;SET UP REGADR
 4043  012330  016103  000016          MOV     SSR(R1),R3      ;SAVE WAS DATA
 4044  012334  004767  010336          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4045  012340  004567  010552          JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
 4046  012344  025144                  LINE
 4047  012346  031563                  EMS7+44
 4048  012350  104057                  ERROR   57              ;READING SILO FAILED TO DEC SSR OR
                                                               ;STORAGE OVFL SET AT WRONG COUNT
 4050  012352  000640                  BR      1$              ;GO TRY NEXT LINE
```

```
4053                                ;*****************************************************************
 (3)                                ;*TEST 41       TRANSMITTER TIMING TEST - ALL LINES - ALL SPEEDS
 (3)                                ;*****************************************************************
 (2)  012354  000004                TST41:  SCOPE
 (1)  012356  012767  000041  166650        MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
4054                                .REM    X
 (1)                                TEST ABSTRACT:
 (1)                                **************
 (1)
 (1)                                        THIS TEST PERFORMA A "RELATIVE" TIMING TEST FOR ALL BAUD RATES
 (1)                                ON ALL SELECTED LINES. IT DOES NOT MEASURE ABSOLUTE TIMES BUT
 (1)                                SIMPLY VERIFIES THAT EACH SUCCESSIVE SPEED FROM 50 TO 9600
 (1)                                BAUD IS FASTER THAN THE PREVIOUS SPEED. THE TEST SEQUENCE IS AS
 (1)                                FOLLOWS:
 (1)
 (1)                                        1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
 (1)                                        2.  INIT "STMP7" TO START WITH 50 BAUD AND A RELATIVE
 (1)                                            TIMER "TIMEC" TO -1 (177777)
 (1)                                        3.  CLEAR THE DH11 AND ACTIVATE SELECTED LINE TO TRANSMIT
 (1)                                            THREE CHARS.
 (1)                                        4.  ACTIVATE TIMER TO UPDATE "TIMEB" THE LINE SPEED TIMER
 (1)                                        5.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR
 (1)                                            AND REPEAT 3 THRU 4 UNTIL ALL SPEEDS CHECKED - THEN
 (1)                                            REPEAT 1 THRU 5 UNTIL ALL LINES CHECKED
 (1)                                        6.  IF "XMIT DONE" SETS VERIFY [TIMEB] LESS THAN [TIMEC]
 (1)                                            IF NOT REPORT ERROR - MAKE [TIMEC]=[TIMEB] AND
 (1)                                            REPEAT 3 THRU 5 UNTIL ALL SPEEDS CHECKED
 (1)                                        7.  REPEAT 1 THRU 6 FOR ALL SELECTED LINES.
 (1)
 (1)                                ERRORS:
 (1)                                *******
 (1)
 (1)                                1.  [ERROR 53] IS CALLED TO REPORT XMIT TIMEOUT ERRORS
 (1)                                2.  [ERROR 17] IS CALLED TO REPORT TIMING ERRORS
 (1)
 (1)                                SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                *****
 (1)
 (1)                                DEBUG:
 (1)                                ******
 (1)
 (1)                                1.  IF ALL LINES FAIL ON ALL SPEEDS SUSPECT THE CLOCK MODULE M4540
 (1)                                2.  IF ALL LINES FAIL ON JUST ONE SPEED (THE SAME ONE) SUPECT EITHER THE
 (1)                                    CLOCK MODULE OR THE M7288 MODULE (TIMING SELECT MUXES)
 (1)                                3.  IF JUST ONE LINE FAILS SUSPECT EITHER THE UART MODULE (M7280)
 (1)                                    EITHER FOR LINES <15:08> OR <07:00> OR THE M7288 MODULE
 (1)
 (1)                                KEY LOGIC:
 (1)                                **********
 (1)
 (1)                                        M4540   SH2             <9600:50> BAUD SIGNALS
 (1)                                        M7288   SH3             BOT AND TOP BUF CLOCK SIGNALS
 (1)                                                SH4,6,8,OR 10   TX CLOCK MAINT SIGNALS
 (1)                                        M7280   TBMT LINE "N" SIGNALS ON UART PIN 22
```

```
                                                TX CLOCK LINE "N" SIGNALS ON UART PIN 40

                                    %
4055  012364  012767  012414  166516            MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4056  012372  004767  010426            1$:    JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
4057  012376  000534                            BR      TST42           ;;BR IF TESTED ALL SELECTED LINES
4058  012400  012767  002100  166610            MOV     #2100,$TMP7     ;INIT TI START WITH LOWEST SPEED
4059  012406  012767  177777  012562            MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
4060  012414  012711  004000            2$:    MOV     #BIT11,(R1)     ;CLEAR THE DH11
4061  012420  156711  012520            3$:    BISB    LINE,(R1)       ;SELECT IT IN THE SCR
4062  012424  012761  177775  000010            MOV     #-3,BCR(R1)     ;SET BYTE COUNT TO XFER 3 CHARS
4063  012432  005061  000006                    CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
4064  012436  016761  166554  000004            MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
4065  012444  016761  011742  000012            MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER
4066
4067  012452  012767  000001  012512            MOV     #1,TIMEA        ;INIT TIMER A
4068  012460  005067  012510                    CLR     TIMEB           ;INIT TIMER B
4069  012464  005711                    4$:    TST     (R1)            ;XMITTR DONE SET YET ?
4070  012466  100437                            BMI     5$              ;BR IF YES
4071  012470  004767  011456                    JSR     PC,TIMEIT       ;CALL THE TIMER
4072  012474  000773                            BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
4073                                                                    ;AROUND THIS BRANCH IF TIME OUT OCCURS
4074
4075  012476  016767  166514  166474            MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
4076  012504  000367  166470                    SWAB    $TMP0
4077  012510  006267  166464                    ASR     $TMP0
4078  012514  006267  166460                    ASR     $TMP0
4079  012520  042767  177760  166452            BIC     #177760,$TMP0
4080  012526  011103                            MOV     (R1),R3         ;GET THE WAS DATA
4081  012530  042703  000200                    BIC     #BIT07,R3       ;CLEAR UNINTERESTING BITS
4082  012534  010102                            MOV     R1,R2           ;MAKE REGADR = DEVADR
4083  012536  012704  100000                    MOV     #BIT15,R4       ;SET UP S/B DATA
4084  012542  156704  012376                    BISB    LINE,R4
4085  012546  004767  010124                    JSR     PC,$UER2A       ;GO SET UP ERROR INFO
4086  012552  004567  010340                    JSR     R5,$UNUM        ;GO SET LINE NO. IN MSG
4087  012556  025144                            LINE
4088  012560  031076                            EMSG+53
4089  012562  104053                            ERROR   53              ;TIMED OUT WAITING FOR XMIT DONE
4090  012564  000426                            BR      8$              ;GO TEST NEXT SPEED
4091
4092  012566  016703  012402            5$:    MOV     TIMEB,R3        ;GET THE WAS COUNT
4093  012572  016704  012400                    MOV     TIMEC,R4        ;GET LASTR CHECK COUNT
4094  012576  020304                            CMP     R3,R4           ;COMPARE RELATIVE TIMES
4095  012600  103420                            BLO     8$              ;BR IF THIS SPEED FASTER THAN LAST
4096                                                                    ;SPEED TESTED
4097
4098  012602  004767  011526            7$:    JSR     PC,$APS         ;SAVE THE ERROR PSW
4099  012606  016702  166404                    MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
4100  012612  000302                            SWAB    R2
4101  012614  006202                            ASR     R2
4102  012616  006202                            ASR     R2
4103  012620  042702  177760                    BIC     #177760,R2      ;STRIP AWAY ALL JUNK
4104  012624  004767  010046                    JSR     PC,$UER2A       ;GO SET UP ERROR INFO
4105  012630  004567  010262                    JSR     R5,$UNUM        ;GO PUT LINE NO. IN MSG
```

```
 4106  012634  025144                          LINE
 4107  012636  026441                          EM17+41
 4108  012640  104017                          ERROR   17              ;TRANSMITTER SPEED INCORRECT
 4109
 4110  012642  016767  012326  012326  8$:     MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
 4111  012650  062767  002100  166340          ADD     #2100,STMP7     ;GENERATE NEXT SPEED
 4112  012656  022767  035600  166332          CMP     #35600,STMP7    ;DONE ALL SPEEDS ?
 4113  012664  001^^3                          BNE     2$              ;BR IF NOT
 4114  012666  000~41                          BR      1$              ;GO TEST NEXT LINE
 4115
```

```
 4118                                   ;*****************************************************************
  (3)                                   ;*TEST 42       RECEIVER TIMING TEST - ALL LINES - ALL SPEEDS
  (3)                                   ;*****************************************************************
  (2)  012670  000004                   TST42:  SCOPE
  (1)  012672  012767  000042  166334           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4119                                   .REM    X
  (1)                                   TEST ABSTRACT:
  (1)                                   ***************
  (1)
  (1)                                           THIS TEST ID IDENTICAL TO TES 40 EXCEPT IT WAITS FOR "DATA READY"
  (1)                                   TO CHECK RECEIVER TIMING. THE SEQUENCE IS SIMILAR AND THE SAME TIMERS
  (1)                                   ARE USED FOR ERROR CHECKING.
  (1)
  (1)                                   ERRORS:
  (1)                                   *******
  (1)
  (1)                                   1.  [ERROR 54] IS CALLED TO REPORT RCVR TIMEOUT ERRORS
  (1)                                   2.  [ERROR 20] IS CALLED TO REPORT RCVR TIMING ERRORS
  (1)
  (1)                                   SYNC:   M7277   SH3     INIT A H        EF2
  (1)                                   *****
  (1)
  (1)                                   DEBUG:          (SAME AS TEST 40)
  (1)                                   ******
  (1)
  (1)                                   KEY LOGIC:      (SAME AS TEST 40 PLUS)
  (1)                                   **********
  (1)
  (1)                                           M7288   SH5,7,9,11       RX CLOCK NN L SIGNALS
  (1)                                           M7280   BUF DH LINE "N" UART PIN 19
  (1)                                                   RX CLOCK LINE "N" UART PIN 17
  (1)
  (1)                                   X
 4120  012700  012767  012730  166202           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 4121  012706  004767  010112          1$:      JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
 4122  012712  000532                           BR      TST43           ;;BR IF TESTED ALL SELECTED LINES
 4123  012714  012767  002100  166274           MOV     #2100,$TMP7     ;INIT TO START WITH LOWEST SPEED
 4124  012722  012767  177777  012246           MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
 4125  012730  012711  004000          2$:      MOV     #BIT11,(R1)     ;CLEAR THE DH11
 4126  012734  156711  012204          3$:      BISB    LINE,(R1)       ;SELECT IT IN THE SCR
 4127  012740  012761  177777  000010           MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO XFER 1 CHAR
 4128  012746  005061  000006                   CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
 4129  012752  016761  166240  000004           MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
 4130  012760  016761  011426  000012           MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER
 4131
 4132  012766  012767  000001  012176           MOV     #1,TIMEA        ;INIT TIMER A
 4133  012774  005067  012174                   CLR     TIMEB           ;INIT TIMER B
 4134  013000  105711                  4$:      TSTB    (R1)            ;RCVR DONE YET ??
 4135  013002  100435                           BMI     5$              ;BR IF YES
 4136  013004  004767  011142                   JSR     PC,TIMEIT       ;CALL THE TIMER
 4137  013010  000773                           BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
 4138                                                                   ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4139
 4140  013012  016767  166200  166160           MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
```

```
 4141  013020  006367  166154                 ASL     $TMPO
 4142  013024  006367  166150                 ASL     $TMPO
 4143  013030  000367  166144                 SWAB    $TMPO
 4144  013034  042767  177760  166136         BIC     #177760,$TMPO
 4145  013042  011103                         MOV     (R1),R3         ;GET THE WAS DATA
 4146  013044  010102                         MOV     R1,R2           ;MAKE REGADR = DEVADR
 4147  013046  012704  100200                 MOV     #BIT15+BIT07,R4 ;SET UP S/B DATA
 4148  013052  156704  012066                 BISB    LINE,R4
 4149  013056  004767  007614                 JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4150  013062  004567  010030                 JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
 4151  013066  025144                         LINE
 4152  013070  026720                         EM22+51
 4153  013072  104054                         ERROR   54              ;TIMED OUT WAITING FOR CHAR AVAIL
 4154  013074  000426                         BR      8$              ;GO TEST NEXT SPEED
 4155
 4156  013076  016703  012072         5$:     MOV     TIMEB,R3        ;GET THE WAS COUNT
 4157  013102  016704  012070                 MOV     TIMEC,R4        ;GET THE CHECK COUNT
 4158  013106  020304                         CMP     R3,R4           ;COMPARE RELATIVE TIMES
 4159  013110  103420                         BLO     8$              ;BR IF TIME INDICATES THIS SPEED FASTER
 4160                                                                 ;THAN LAST SPEED
 4161
 4162  013112  004767  011216         7$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4163  013116  016702  166074                 MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
 4164  013122  006302                         ASL     R2
 4165  013124  006302                         ASL     R2
 4166  013126  000302                         SWAB    R2
 4167  013130  042702  177760                 BIC     #177760,R2      ;STRIP AWAY ALL JUNK
 4168  013134  004767  007536                 JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4169  013140  004567  007752                 JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
 4170  013144  025144                         LINE
 4171  013146  026600                         EM20+36
 4172  013150  104020                         ERROR   20              ;RECEIVER SPEED INCORRECT
 4173
 4174  013152  016767  012016  012016  8$:     MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
 4175  013160  062767  002100  166030         ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
 4176  013166  022767  035600  166022         CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
 4177  013174  001255                         BNE     2$              ;BR IF NOT
 4178  013176  000643                         BR      1$              ;GO TEST NEXT LINE
 4179
```

```
4182                                    ;*****************************************************************
 (3)                                    ;*TEST 43       BASIC DATA TEST - ALL LINES/ALL CHAR LENGTHS
 (3)                                    ;*****************************************************************
 (2)  013200  000004                    TST43:  SCOPE
 (1)  013202  012767  000043  166024            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
4183                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT A SINGLE ALL ONES CHAR. CAN BE TRANS-
 (1)                                    MITTED AND RECEIVED ON ALL SELECTED LINES AT ALL FOUR CHAR LENGTHS
 (1)                                    (5, 6, 7, AND 8 BITS). THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SET UP THE ERROR LOOP RETURN
 (1)                                            2. SELECT A LINE NO. TO TEST - GO TO TEST 43 IF DONE ALL
 (1)                                               SELECTED LINES.
 (1)                                            3. GET A TEST CHARACTER FROM THE DATA TABLE AND UPDATE THE
 (1)                                               TABLE POINTER.
 (1)                                            4. CLEAR THE DH11
 (1)                                            5. PRIME THE SELECTED LINE TO XMIT ONE CHAR AT 9600 BAUD
 (1)                                            6. WAIT FOR "CHAR AVAIL" TO SET - IF TIMEOUT REPORT ERROR AND
 (1)                                               RESTART AT STEP 8.
 (1)                                            7. IF NO TIMEOUT - CHECK DATA AND REPORT ANY ERRORS
 (1)                                            8. INCREMENT "SCR" REG TO CHANGE CHAR LENGTH - IF DONE
 (1)                                               ALL FOUR GO TO STEP 2 - IF NOT THEN STEP 4.
 (1)
 (1)                                    ERRORS:
 (1)                                    ******
 (1)
 (1)                                    1. [ERROR  55] IS CALLED TO REPORT RCVR TIMEOUT.
 (1)                                    2. [ERROR  23] IS CALLED TO REPORT DATA COMPARE ERRORS
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1. IF FAULT AFFECTS ONLY ONE LINE AT ALL CHAR LENGTHS, SUSPECT A
 (1)                                       BAD UART MODULE M7280.
 (1)
 (1)                                    2. IF FAULT AFFECTS ONLY ONE BIT ON ALL LINES, SUSPECT THE
 (1)                                        THE M7279 MODULE.
 (1)                                    3. IF FAULT AFFECTS ONLY CERTAIN CHAR LENGTHS, SUSPECT EITHER THE M7278
 (1)                                       OR THE UART MODULE M7280.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7280'S UART CHIPS PINS <12:05>
 (1)
 (1)                                            M7279   SH1     E1,E2,E6, OR E7
 (1)
 (1)                                            M7278   SH8     N82 LPR 01 H    FH1
```

```
(1)                                                    MB1 LPR 00 H    FH?
(1)                                 %
4184  013210  012767  013236  165672          MOV     #3$,SLPERR      ;SET UP ERRVR LOOP RETURN
4185  013216  004767  007602              1$:  JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4186  013222  000511                          BR      TST44           ;BR IF DONE ALL SELECTED LINES
4187  013224  012705  025160                  MOV     #TDATA2,R5      ;GET POINTER TO DATA TABLE
4188  013230  005002                          CLR     R2              ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
4189  013232  012567  165760              2$: MOV     (R5)+,STMP7     ;PUT TEST CHAR IN XMIT BUFFER
4190  013236  012711  004000              3$: MOV     #BIT11,(R1)     ;CLEAR THE DH11
4191  013242  156711  011676                  BISB    LINE,(R1)       ;SELECT THE LINE
4192  013246  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
4193  013254  012761  001216  000006          MOV     #STMP7,CAR(R1)  ;SET CURRENT ADDRESS REG
4194  013262  012761  033500  000004          MOV     #33500,LPR(R1)  ;SET BAUD RATE TO 9600
4195  013270  050261  000004                  BIS     R2,LPR(R1)      ;SELECT CHAR LENGTH
4196  013274  156761  011112  000012          BISB    LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
4197
4198  013302  012767  000001  011662          MOV     #1,TIMEA        ;INIT TIMER A
4199  013310  005067  011660                  CLR     TIMEB           ;INIT TIMER B
4200  013314  105711                      4$: TSTB    (R1)            ;RCVR DONE YET ??
4201  013316  100424                          BMI     5$              ;BR IF YES
4202  013320  004767  010626                  JSR     PC,TIMEIT       ;CALL THE TIMER
4203  013324  000773                          BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
4204                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4205
4206  013326  004767  011002                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4207  013332  011103                          MOV     (R1),R3         ;GET THE SCR
4208  013334  042703  177560                  BIC     #177560,R3      ;CLEAR UNINTERESTING BITS
4209  013340  012704  000200                  MOV     #200,R4         ;SET UP S/B DATA
4210  013344  156704  011574                  BISB    LINE,R4
4211  013350  004767  007322                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4212  013354  004567  007536                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
4213  013360  025144                          LINE
4214  013362  026720                          EM22+51
4215  013364  104055                          ERROR   55              ;CHAR AVAIL FAILED TO SET ON TIME
4216  013366  000422                          BR      6$              ;GO TEST NEXT CHAR LENGTH
4217
4218  013370  016103  000002              5$: MOV     NRC(R1),R3      ;GET THE WAS DATA
4219  013374  012704  000200                  MOV     #200,R4         ;SET UP THE S/B DATA IN R4
4220  013400  156704  011540                  BISB    LINE,R4
4221  013404  000304                          SWAB    R4
4222  013406  156704  165604                  BISB    STMP7,R4
4223  013412  020304                          CMP     R3,R4           ;WAS THE RCVD DATA CORRECT ??
4224  013414  001407                          BEQ     6$              ;BR IF YES
4225
4226  013416  004767  007250                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
4227  013422  004567  007470                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
4228  013426  025144                          LINE
4229  013430  026761                          EM23+36
4230  013432  104023                          ERROR   23              ;DATA COMPARE ERROR
4231
4232  013434  005202                      6$: INC     R2              ;DO NEXT CHAR LENGTH ON SELECTED LINE
4233  013436  022702  000004                  CMP     #4,R2           ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
4234  013442  001273                          BNE     2$              ;BR IF NOT
4235  013444  000664                          BR      1$              ;GO DO NEXT LINE
```

```
                                ;*****************************************************************
                                ;*TEST 44       SINGLE LINE DATA TEST - ALL LINES
                                ;*****************************************************************
013446  000004                  TST44:  SCOPE
013450  012767  000044  165556          MOV     #STN-1,STESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST TRANSMITS AND RECEIVES A BINARY COUNT PATTERN
                                (000 - 377) ON ALL SELECTED LINES. THE TEST SEQUENCE IS AS
                                FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. GO SELECT A LINE NO. TO TEST - IF DONE ALL SELECTED
                                           LINES THEN GO TO TEST 44.
                                        3. CLEAR THE DH11 AND PRIME THE SELECTED LINE TO XMIT
                                           TO XMIT 256. CHARS AT 9600. BAUD - 8 BIT CHARS.
                                        4. SET UP RS TO POINT TO RCVR CORE BUFFER.
                                        5. ACTIVATE THE SELECTED XMITTER.
                                        6. WAIT FOR "CHAR AVAIL" TO SET BEFORE READING THE SILO.
                                           IF RCVR TIMEOUT REPORT ERROR AND RESTART AT STEP 2.
                                        7. IF NO TIMEOUT READ THE SILO AND STORE THE WORD IN THE RCVR
                                           CORE BUFFER - WHEN THE BUFFER IS FULL GO TO STEP8 IF NOT
                                           THEN GO TO STEP 6.
                                        8. COMPARE THE XMIT AND RCVR CORE IMAGE BUFFERS AND REPORT
                                           ALL DATA COMPARE ERRORS.
                                        9. CHECK THE "BAR", "BCR", AND "CAR" REGISTERS FOR CORRECT
                                           CONTENTS - REPORT ALL ERRORS.
                                        10.GO TO STEP 2

                                                NOTE:   THIS TEST USES THE MAINT. BIT (SCRD9=1) TO
                                                        TURN THE DATA AROUND INSTEAD OF THE MAINT
                                                        CONNECTORS OR MODULE.

                                ERRORS:
                                *******

                                1. [ERROR  22] IS CALLED TO REPORT "DATA AVAIL" TIMEOUT
                                2. [ERROR  37]  "        "          " DATA COMPARE ERRORS
                                3. [ERROR  40]  "        "          " "BAR" REG NOT CLEARED
                                4. [ERROR  10]  "        "          " "BCR" REG NOT ALL ZEROES
                                5. [ERROR   7]  "        "          " "CAR" REG NOT UPDATED CORRECTLY

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                *******

                                1. IF THE FAULT AFFECTS ONE OR MORE LINES IN AN 8 LINE GROUP <15:08>
                                   OR <07:00>, SWAP THE M7280 MODULES. IF THE FAULT SHIFTS SO THAT
                                   THE ERROR INDICATES DIFFERENT LINES THE PROBLEM IS MOST LIKELY
                                   THE M7280 THE SYMPTOM SHIFTED TO.
```

```
(1)
(1)                                     2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON
(1)                                        ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.
(1)
(1)                                     3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS
(1)                                        ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.
(1)
(1)                                     4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS
(1)                                        SUSPECT THE M7278 OR M7277 MODULES RESPECTFULLY.
(1)
(1)                                     KEY LOGIC       (REFER TO TEST 43)
(1)                                     ***********
(1)
(1)                                     %
4240  013456  012767  013500  165424          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4241  013464  004767  007334          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4242  013470  000401                          BR      11$             ;;BR IF ALL SELECTED LINES DONE
4243  013472  000402                          BR      2$              ;GO TEST IT
4244  013474  000167  000424          11$:    JMP     8$              ;EXIT TEST
4245  013500  016701  010674          2$:     MOV     DHADR,R1        ;RESET DEVADR
4246  013504  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
4247  013510  156711  011430                  BISB    LINE,(R1)       ;SET SELECT BITS IN SCR
4248  013514  012761  033436  000006          MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG
4249  013522  012761  177400  000010          MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT
4250  013530  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET LINE PARAMETERS
4251  013536  012705  032436                  MOV     #RBUF,R5        ;SET UP POINTER TO INPUT DATA BUFFER
4252  013542  052711  001000                  BIS     #BIT09,(R1)     ;SET MAINT MODE BIT
4253  013546  016761  010640  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
4254
4255  013554  012767  000002  011410          MOV     #2,TIMEA        ;INIT TIMER A
4256  013562  005067  011406                  CLR     TIMEB           ;INIT TIMER B
4257  013566  105711                  3$:     TSTB    (R1)            ;RCVR DONE YET ??
4258  013570  100425                          BMI     4$              ;BR IF YES
4259  013572  004767  010354                  JSR     PC,TIMEIT       ;CALL THE TIMER
4260  013576  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4261                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4262
4263  013600  004767  010530                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4264  013604  010102                          MOV     R1,R2           ;SET UP REGADR
4265  013606  011203                          MOV     (R2),R3         ;GET THE WAS DATA
4266  013610  042703  176400                  BIC     #176400,R3      ;CLEAR JUNK BITS
4267  013614  012704  001200                  MOV     #1200,R4        ;SET UP S/B DATA
4268  013620  156704  011320                  BISB    LINE,R4
4269  013624  004767  007046                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4270  013630  004567  007262                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4271  013634  025144                          LINE
4272  013636  026720                          EM22+51
4273  013640  104022                          ERROR   22              ;CHAR AVAIL TIMEOUT
4274  013642  000710                          BR      1$              ;GO TRY NEXT LINE
4275
4276  013644  016125  000002          4$:     MOV     NRC(R1),(R5)+   ;SAVE THE RECEIVED DATA
4277  013650  022705  033436                  CMP     #RBUF+1000,R5   ;INPUT BUFFER FULL ??
4278  013654  001344                          BNE     3$              ;BR IF NOT
4279
```

```
4280  013656  012702  033436          MOV     #TBUF,R2        ;SET UP POINTER TO OUTPUT BUFFER
4281  013662  012701  032436          MOV     #RBUF,R1        ;SET UP POINTER TO INPUT BUFFER
4282  013666  111204              5$:  MOVB    (R2),R4         ;SET UP S/B DATA IN R4
4283  013670  042704  177400          BIC     #177400,R4
4284  013674  000304                  SWAB    R4
4285  013676  156704  011242          BISB    LINE,R4
4286  013702  152704  000200          BISB    #200,R4
4287  013706  000304                  SWAB    R4
4288  013710  011103                  MOV     (R1),R3         ;GET THE WAS DATA
4289  013712  020304                  CMP     R3,R4           ;DATA CORRECT ??
4290  013714  001407                  BEQ     6$              ;BR IF YES
4291
4292  013716  004767  006750          JSR     PC,SUER2        ;GO SET UP ERROR INFO
4293  013722  004567  007170          JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4294  013726  025144                  LINE
4295  013730  030176                  EM37+33
4296  013732  104037                  ERROR   37              ;DATA COMPARE ERROR
4297
4298
4299  013734  005202              6$: INC     R2              ;UPDATE DATA BUFFER POINTERS
4300  013736  062701  000002          ADD     #2,R1           ;
4301  013742  022701  033436          CMP     #RBUF+1000,R1   ;COMPARED ALL 256. CHARS ??
4302  013746  001347                  BNE     5$              ;BR IF NOT
4303
4304  013750  016701  010424          MOV     DHADR,R1        ;RESET DEVADR
4305  013754  010102                  MOV     R1,R2           ;SET UP REGADR
4306  013756  062702  000012          ADD     #BAR,R2
4307  013762  005712                  TST     (R2)            ;WAS THE "BAR" ALL ZEROES ??
4308  013764  001413                  BEQ     7$              ;BR IF YES
4309
4310  013766  004767  010342          JSR     PC,SAPS         ;SAVE THE ERROR PSW
4311  013772  011203                  MOV     (R2),R3         ;GET THE WAS DATA
4312  013774  005004                  CLR     R4              ;SET UP S/B DATA
4313  013776  004767  006674          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4314  014002  004567  007110          JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4315  014006  025144                  LINE
4316  014010  030241                  EM40+40
4317  014012  104040                  ERROR   40              ;"BAR" REG NOT ALL ZEROES
4318
4319  014014  010102              7$: MOV     R1,R2           ;SET UP REGADR
4320  014016  062702  000010          ADD     #BCR,R2
4321  014022  005712                  TST     (R2)            ;BYTE COUNT REG ALL ZEROES ?
4322  014024  001413                  BEQ     71$             ;;BR IF BYTE COUNT ZERO
4323
4324  014026  004767  010302          JSR     PC,SAPS         ;SAVE THE ERROR PSW
4325  014032  011203                  MOV     (R2),R3         ;GET THE WAS DATA
4326  014034  005004                  CLR     R4              ;SET UP THE S/B DATA
4327  014036  004767  006634          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4328  014042  004567  007050          JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4329  014046  025144                  LINE
4330  014050  026046                  EM10+44
4331  014052  104010                  ERROR   10              ;BYTE COUNT NOT ALL ZEROES
4332
4333  014054  010102             71$: MOV     R1,R2           ;SET UP REGADR
```

```
4334  014056  062702  000006              ADD     #CAR,R2
4335  014062  022712  034036              CMP     #TBUF+400,(R2)  ;DID "CAR" INCREMENT PROPERLY ?
4336  014066  001414                      BEQ     72$             ;BR IF YES
4337
4338  014070  004767  010240              JSR     PC,SAPS         ;SAVE THE ERROR PSW
4339  014074  011203                      MOV     (R2),R3         ;GET THE WAS DATA
4340  014076  012704  034036              MOV     #TBUF+400,R4    ;SET UP S/B DATA
4341  014102  004767  006570              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4342  014106  004567  007004              JSR     R5,SUNUM        ;GO PUT LINE NO IN MESSAGE
4343  014112  025144                      LINE
4344  014114  025777                      EM7+47
4345  014116  104007                      ERROR   7               ;"CAR" NOT UPDATED CORRECTLY
4346
4347  014120  000167  177340      72$:    JMP     1$              ;GO DO NEXT LINE
4348
4349  014124  000240              8$:     NOP                     ;EXIT POINT
```

```
                                ;*****************************************************************
                                ;*TEST 45       BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY
                                ;*****************************************************************
014126  000004                  TST45:  SCOPE
014130  012767  000045  165076          MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    X
                                TEST ABSTRACT:
                                ****************

                                        THIS TEST VERIFIES THE ODD PARITY FUNCTION FOR ALL SELECTED
                                LINES USING THE "BREAK" FUNCTION TO FORCE PARITY ERRORS. REFER TO
                                THE FLOW CHARTS IN THE PROGRAM DOCUMENTATION FOR TEST SEQUENCES.

                                ERRORS:
                                *******

                                1. [ERROR  22] IS CALLED TO REPORT RCVR TIMEOUT
                                2. [ERROR  33] IS CALLED TO REPORT DATA/PARITY ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1. IF FAULT AFFECTS ALL LINES SUSPECT THE M7278 MODULE.

                                2. IF IT AFFECTS ONLY ONE LINE SUSPECT THE "UART" MODULE FOR THAT LINE.


                                KEY LOGIC:
                                **********

                                        M7278   SH7     PEN LPR04 L     FF2
                                                        PEV LPR05 L     FN1
                                X
014136  012767  011152  164744          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
014144  004767  006654          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE NO.
014150  000506                  BR      TST46                   ;;BR IF ALL SELECTED LINES DONE
014152  012711  004000          2$:     MOV     #DIT11,(R1)     ;CLEAR OUT THE DH11
014156  012704  000260                  MOV     #260,R4         ;SET UP THE S/B DATA IN R4
014162  156704  010756                  BISB    LINE,R4
014166  000304                  SWAB    R4
014170  156711  010750                  BISB    LINE,(R1)       ;SET LINE NO. IN SCR
014174  012767  000377  010754          MOV     #377,TDATA1     ;LOAD XMIT BUFFER WITH TEST CHARACTER
014202  012761  073563  000004          MOV     #73563,LPR(R1)  ;SET UP THE LINE PARAMETERS
014210  012761  177777  000010          MOV     #-1,BCR(R1)     ;LOAD THE BYTE COUNT REG
014216  012761  025156  000006          MOV     #TDATA1,CAR(R1) ;LOAD THE BUS ADDR REG
014224  016761  010162  000014          MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR SELECTED LINE
014232  016761  010154  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE XMITTR

014240  012767  000001  010724          MOV     #1,TIMEA        ;INIT TIMER A
014246  005067  010722                  CLR     TIMEB           ;INIT TIMER B
014252  105711                  3$:     TSTB    (R1)            ;RCVR DONE YET ??
```

```
4373  014254  100423                        BMI     4$              ;BR IF YES
4374  014256  004767  007670                JSR     PC,TIMEIT       ;CALL THE TIMER
4375  014262  000773                        BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4376                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4377
4378  014264  004767  010044                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4379  014270  011103                        MOV     (R1),R3         ;GET THE WAS DATA
4380  014272  012704  100200                MOV     #100200,R4      ;SET UP THE S/B DATA
4381  014276  156704  010642                BISB    LINE,R4
4382  014302  010102                        MOV     R1,R2           ;SET UP REGADR
4383  014304  004767  006366                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4384  014310  004567  006602                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4385  014314  025144                        LINE
4386  014316  026720                        EM22+51
4387  014320  104022                        ERROR   22              ;TIMED OUT WAITING FOR DATA AVAIL
4388  014322  000710                        BR      1$              ;GO TEST NEXT LINE
4389
4390  014324  016103  000002        4$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
4391  014330  020304                        CMP     R3,R4           ;CORRECT DATA RECEIVED ??
4392  014332  001704                        BEQ     1$              ;BR IF YES
4393
4394  014334  004767  007774                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4395  014340  010102                        MOV     R1,R2           ;SET UP THE REGADR
4396  014342  062702  000002                ADD     #NRC,R2
4397  014346  004767  006324                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4398  014352  004567  006540                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4399  014356  025144                        LINE
4400  014360  027712                        EM33+40
4401  014362  104033                        ERROR   33              ;INCORRECT DATA OR PARITY ERROR
4402  014364  000667                        BR      1$              ;GO TEST NEXT LINE
```

```
                                ;*********************************************************
                                ;*TEST 46       MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
                                ;*********************************************************
014366  000004                  TST46:  SCOPE
014370  012767  000046  164636          MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    X
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE
                                A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7,
                                AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED.
                                THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH
                                SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB:".  REFER TO THE FLOW
                                CHARTS IN THE PROGRAM DOCUMENT FOR A DETAILED DESCRIPTION OF THE TEST
                                SEQUENCES.

                                ERRORS:
                                *******

                                1. [ERROR  41] IS CALLED TO REPORT FALSE RECEIVER INTRS.
                                2. [ERROR  42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
                                3. [ERROR  34] IS CALLED TO REPORT PARITY/DATA ERRORS
                                4. [ERROR  35] IS CALLED TO REPORT TEST TIMEOUTS

                                SYNC:   (NONE)
                                *****

                                DEBUG:  (REFER TO TEST 45)
                                ******

                                KEY LOGIC:      (REFER TO TEST 45)
                                **********

                                %
014376  012767  014414  164504           MOV     #1$,SLPERR      ;SET UP THE ERROR LOOP RETURN
014404  012705  024462                   MOV     #PRTYTB+4,R5    ;SET UP POINTER TO TEST PARAMETERS
014410  005067  164602                   CLR     STMP7           ;START WITH SUB TEST #00
014414  162705  000004           1$:     SUB     #4,R5           ;RESET POINTER FOR ERROR LOOPS
014420  005367  164572                   DEC     STMP7           ;RESET SUB TEST # FOR ERROR LOOP
014424  022705  024516           2$:     CMP     #PRTYTB+40,R5   ;DONE ALL 8. SUB TESTS ??
014430  001456                           BEQ     21$             ;BR IF YES
014432  012706  001100                   MOV     #STACK,SP       ;RESET STACK POINTER FOR ERROR LOOPS
014436  016701  007736                   MOV     DHRDR,R1        ;RESET DEVADR FOR ERROR LOOPS
014442  012567  164546                   MOV     (R5)+,STMP4     ;GET THE BYTE COUNT PARAMETER
014446  012567  164540                   MOV     (R5)+,STMP5     ;GET THE LINE PARAMETERS
014452  005267  164540                   INC     STMP7           ;GENERATE NEW SUB-TEST NO.
014456  012711  004000                   MOV     #BIT11,(R1)     ;CLEAR THE DHU
014462  004767  006612                   JSR     PC,SUPPAR       ;GO SET UP PARAMETERS
014466  016167  000004  164504           MOV     LPR(R1),STMP0   ;SAVE CURRENT LINE PARAMETERS
014474  016767  007710  164504           MOV     LINSEL,STMP3    ;SAVE SELECTED LINES PARAMETER
014502  004767  007612                   JSR     PC,CHPS2        ;GO LOCK OUT INTRS
014506  016702  007670                   MOV     DHVCT,R2        ;SET UP THE VECTOR
014512  012722  014604                   MOV     #3$,(R2)+       ;GO TO 3$ ON RCVR INTERRUPT
```

```
4426  014516  116712  010416                MOVB    DHRLVL,(R2)
4427  014522  012711  000100                MOV     #100,(R1)       ;ENABLE CHAR AVAIL INTERRUPTS
4428  014526  016767  007656  010122        MOV     LINSEL,LINACT   ;FLAG ALL SELECTED LINES ACTIVE
4429  014534  016761  007650  000012        MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTEDLINES
4430  014542  116767  164334  164434        MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
4431  014550  042767  177400  164426        BIC     #177400,$TMP2
4432  014556  004767  007522                JSR     PC,CHPS1        ;GO CLEAR PSW
4433  014562  000167  000176                JMP     7$              ;GO WAIT FOR INTERRUPTS
4434
4435  014566  012706  001100        21$:    MOV     #STACK,SP       ;RESTORE THE SP
4436  014572  004767  007506                JSR     PC,CHPS1        ;GO CLEAR PSW
4437  014576  004767  007322                JSR     PC,RESTRP       ;RESTORE TRAP CATCHER
4438  014602  000536                        BR      TST47           ;;GO TO NEXT TEST
4439
4440                                ;RECEIVER INTERRUPT SERVICE ROUTINE
4441
4442  014604  105711                3$:     TSTB    (R1)            ;CHAR AVAIL SET
4443  014606  100404                        BMI     4$              ;BR IF YES
4444
4445  014610  012711  004000                MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4446  014614  104041                        ERROR   41              ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
4447  014616  000702                        BR      2$              ;GO TRY NEXT SUB TEST
4448
4449  014620  032711  040000        4$:     BIT     #BIT14,(R1)     ;SILO OVERFLOW ??
4450  014624  001404                        BEQ     5$              ;BR IF NOT
4451
4452  014626  012711  004000                MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4453  014632  104042                        ERROR   42              ;SILO OVERFLOW ERROR
4454  014634  000673                        BR      2$              ;GO TRY NEXT SUB TEST
4455
4456  014636  016103  000002        5$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
4457  014642  010302                        MOV     R3,R2           ;EXTRACT AND SAVE LINE NO.
4458  014644  000302                        SWAB    R2
4459  014646  042702  177760                BIC     #177760,R2
4460  014652  010267  164332                MOV     R2,$TMP4
4461  014656  006302                        ASL     R2              ;GENERATE TABLE OFFSETR
4462  014660  026203  032436                CMP     RBUF(R2),R3     ;CORRECT DATA RECEIVED ??
4463  014664  001426                        BEQ     6$              ;BR IF YES
4464
4465  014666  004767  007442                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4466  014672  012711  004000                MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4467  014676  016204  032436                MOV     RBUF(R2),R4     ;SET UP S/B DATA
4468  014702  062701  000002                ADD     #NRC,R1         ;SET UP WAS ADDRESS
4469  014706  062702  032436                ADD     #RBUF,R2        ;SET UP S/B ADDRESS
4470  014712  004767  005760                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4471  014716  004567  006174                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4472  014722  001210                        $TMP4
4473  014724  027766                        EM34+51
4474  014726  004567  006164                JSR     R5,SUNUM        ;PUT SUBTEST NO. IN MESSAGE
4475  014732  001216                        $TMP7
4476  014734  030004                        EM34+67
4477  014736  104034                        ERROR   34              ;PARITY DATA COMPARE ERROR
4478  014740  000631                        BR      2$              ;GO TRY NEXT SUBTEST
4479
```

```
4480  014742  105262  032436          6$:     INCB    RBUF(R2)        ;GENERATE NEW RCVD DATA
4481  014746  005262  024660                  INC     MULPTB(R2)      ;COUNT ONE BYTES RECEIVED
4482  014752  001003                          BNE     61$             ;BR IF NOT DONE
4483  014754  046267  024616  007E7?          BIC     LINBIT(R2),LINACT ;FLAG THIS LINE DONE
4484  014762  000002                  61$:    RTI                     ;RETURN TO WAIT ROUTINE
4485
4486                                  ;WAIT ROUTINE
4487
4488  014764  012767  000002  010200  7$:     MOV     #2,TIMEA        ;INIT TIMER A
4489  014772  005067  010176                  CLR     TIMEB           ;INIT TIMER B
4490  014776  005761  000012          8$:     TST     BAR(R1)         ;ALL LINES DONE XMITTING ??
4491  015002  001413                          BEQ     9$              ;BR IF YES
4492  015004  004767  007142                  JSR     PC,TIMEIT       ;CALL THE TIMER
4493  015010  000772                          BR      8$              ;TIMER ROUTINE WILL MOVE RETURN PC
4494                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4495
4496  015012  016167  000012  164166          MOV     BAR(R1),$TMP3   ;SAVE THE ACTIVE LINES FLAG
4497  015020  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4498  015024  104035                          ERROR   35              ;TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
4499  015026  000167  177372                  JMP     2$              ;GO TRY NEXT SUBTEST
4500
4501
4502  015032  012767  000001  010132  9$:     MOV     #1,TIMEA        ;INIT TIMER A
4503  015040  005067  010130                  CLR     TIMEB           ;INIT TIMER B
4504  015044  005767  007606          10$:    TST     LINACT          ;ALL CHARS RECEIVED ?
4505  015050  001411                          BEQ     11$             ;BR IF YES
4506  015052  004767  007074                  JSR     PC,TIMEIT       ;CALL THE TIMER
4507  015056  000772                          BR      10$             ;TIMER ROUTINE WILL MOVE RETURN PC
4508                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4509
4510  015060  016767  007572  164120          MOV     LINACT,$TMP3    ;SET UP ACTIVE LINE PARAMETER
4511  015066  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4512  015072  104035                          ERROR   35              ;SILO EMPTY TIMEOUT
4513  015074  000167  177324          11$:    JMP     2$              ;GO TRY NEXT SUB TEST
```

```
 4515                                ;*****************************************************************
  (3)                                ;*TEST 47       AUTO ECHO TEST 1 - ALL LINES
  (3)                                ;*****************************************************************
  (2)  015100  000004                TST47:  SCOPE
  (1)  015102  012767  000047  164124        MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4516                                .REM    X
  (1)                                TEST ABSTRACT:
  (1)                                **************
  (1)
  (1)                                        THIS TEST VERIFIES THAT ALL SELECTED LINES CAN TURN AROUND
  (1)                                A SINGLE TEST CHARACTER (377) AND (000) IN AUTO ECHO MODE. THE TEST SEQUENCE
  (1)                                IS AS FOLLOWS:
  (1)
  (1)                                        1. SET UP THE ERROR LOOP RETURN
  (1)                                        2. RETRIEVE THE AUTO-ECHO TEST DATA FROM "AETAB:" AND
  (1)                                           UPDATE THE POINTER.
  (1)                                        3. GO SELECT A LINE NO. TO TEST - GO TO STEP 10 IF DONE
  (1)                                           ALL SELECTED LINES.
  (1)                                        4. CLEAR THE "CAR" AND "BCR" MEMORIES.
  (1)                                        5. PRIME THE SELECTED LINE TO XMIT ONE CHAR WITH A.E. ENABLED.
  (1)                                        6. ACTIVATE THE SELECTED TRANSMITTER.
  (1)                                        7. WAIT FOR "CHAR AVAIL" - IF TIMEOUT REPORT ERROR AND RESTART
  (1)                                           AT STEP 2
  (1)                                        8. IF NO TIMEOUT - READ SILO AND COMPARE AUTO ECHO DATA RECEIVED
  (1)                                           REPORT DATA COMPARE ERRORS AND RESTART AT STEP 2
  (1)                                        9. IF NO ERRORS REPEAT STEPS 7 AND 8 SIXTY-FOUR TIMES THEN TURN
  (1)                                           OFF A.E. ENABLE AND READ LAST CHAR FROM SILO.. CHECK LAST
  (1)                                           CHAR FOR DATA COMPARE ERRORS - REPORT ERRORS IF ANY - AND RE-
  (1)                                           START AT STEP 2.
  (1)                                       10. CHANGE A.E. TABLE POINTER TO POINT TO "AETAB0:" (0'S DATA)
  (1)                                           AND REPEAT STEPS 2 THRU 9.
  (1)
  (1)                                ERRORS:
  (1)                                ********
  (1)
  (1)                                1. [ERROR 24] IS CALLED TO REPORT ALL ERRORS
  (1)
  (1)                                SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
  (1)                                *****
  (1)
  (1)                                DEBUG:
  (1)                                ******
  (1)
  (1)                                1. IF ALL LINES FAIL, SUSPECT EITHER THE M7277 OR M7289
  (1)
  (1)                                2. IF ONLY ONE LINE FAILS SUSPECT THE M7288
  (1)
  (1)                                3. LOOP ON THE FAILING LINE AND TRACK BACK THROUGH THE KEY LOGIC.
  (1)
  (1)
  (1)                                KEY LOGIC:
  (1)                                **********
  (1)
  (1)                                       M7277   SH3     AE GO L         CS2
```

```
(1)                                                    7402 "OR" GATE CHIPS E38 OR E41
(1)                                                    74157 MUX CHIPS  E39 OR 342
(1)                                            SH4     E35 - PIN 2 STUCK LOW
(1)
(1)                                            M7289   SH3     AE GO L          EK1
(1)                                                    AE SCAN MUX  E22 PIN 10
(1)                                            SH4     SAMPLE STATUS H E21-12
(1)
(1)                                            M7288   SH5, 7, 9, 11
(1)                                                    AE ENABLE "NN" H CONTROL FLOPS
(1)                                                    74174 CHIPS PIN 15
(1)
(1)                                    %
4517  015110  012767  015172  163772          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4518  015116  005067  164064                  CLR     $TMP3           ;INIT I/O DATA FLAG
4519  015122  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
4520  015126  012705  024516                  MOV     #AETAB,R5       ;GET POINTER TO AUTO ECHO DATA TABLE
4521  015132  005067  007256          7$:     CLR     LMSK1           ;INIT BIT TEST MARKER
4522  015136  000261                          SEC                     ;SET "C" BIT FOR MARKER
4523  015140  000401                          BR      13$             ;GO SHIFT MASK
4524  015142  000241                  1$:     CLC                     ;INIT THE "C" BIT
4525  015144  006167  007244          13$:    ROL     LMSK1           ;SHIFT BIT MARKER
4526  015150  001407                          BEQ     12$             ;BR IF DONE ALL LINES
4527  015152  012504                          MOV     (R5)+,R4        ;SET UP THE S/B DATA
4528  015154  036767  007234  007226          BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
4529  015162  001767                          BEQ     1$              ;BR IF NOT
4530  015164  004767  005634          11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4531  015170                          12$:
(1)   015170  000522                          BR      6$              ;;BR IF ALL SELECTED LINES TESTED
4532  015172  004767  006000          2$:     JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
4533  015176  116711  007742                  MOVB    LINE,(R1)       ;SET SELECT BITS IN SCR REG
4534  015202  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET UP TO XFER ONE CHAR
4535  015210  010561  000006                  MOV     R5,CAR(R1)      ;SET UP THE BUS ADDRESS REG
4536  015214  162761  000002  000006          SUB     #2,CAR(R1)      ;CORRECT BUS ADDRESS
4537  015222  012767  000100  163766          MOV     #100,$TMP7      ;COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
4538  015230  005067  163760                  CLR     $TMP6           ;INIT CHAR COUNTER
4539  015234  012761  133503  000004          MOV     #133503,LPR(R1) ;SET UP LINE PARAMETER REG
4540  015242  016761  007144  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE LINE
4541
4542  015250  012767  000002  007714          MOV     #2,TIMEA        ;INIT TIMER A
4543  015256  005067  007712                  CLR     TIMEB           ;INIT TIMERB
4544  015262  105711                  3$:     TSTB    (R1)            ;CHAR AVAIL SET ??
4545  015264  100427                          BMI     4$              ;BR IF YES
4546  015266  004767  006660                  JSR     PC,TIMEIT       ;CALL THE TIMER
4547  015272  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4548                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4549
4550  015274  004767  007034                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4551  015300  005061  000004                  CLR     LPR(R1)         ;TURN OFF AUTO ECHO MODE
4552  015304  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
4553  015306  011103                          MOV     (R1),R3         ;GET THE WAS DATA
4554  015310  042703  100000                  BIC     #BIT15,R3       ;CLEAR JUNK BIT
4555  015314  012704  000200                  MOV     #200,R4         ;SET UP S/B DATA
4556  015320  156704  007620                  BISB    LINE,R4
```

```
4557  015324  004767  005346                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4558  015330  004567  005562                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
4559  015334  025144                          LINE
4560  015336  027116                          EM24+35
4561  015340  104024                          ERROR   24              ;DATA AVAIL FAILED TO SET ON TIME
4562  015342  000677                          BR      1$              ;GO TRY NEXT LINE
4563
4564  015344  005267  163644          4$:     INC     $TMP6           ;COUNT ONE CHAR RECVD
4565  015350  016103  000002                  MOV     NRC(R1),R3      ;GET THE WAS DATA
4566  015354  020304                          CMP     R3,R4           ;WAS CHAR AUTO ECHOED CORRECTLY ?
4567  015356  001417                          BEQ     5$              ;BR IF YES
4568
4569  015360  004767  006750                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4570  015364  005061  000004                  CLR     LPR(R1)         ;DISABLE AUTO ECHO
4571  015370  010102                          MOV     R1,R2           ;SET UP REGADR
4572  015372  062702  000002                  ADD     #NRC,R2
4573  015376  004767  005274                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4574  015402  004567  005510                  JSR     R5,SUNUM        ;PUT LINE NO. IN ERROR MSG
4575  015406  025144                          LINE
4576  015410  027116                          EM24+35
4577  015412  104024                          ERROR   24              ;CHAR AUTO ECHOED INCORRECTLY
4578  015414  000652                          BR      1$              ;GO TRY NEXT LINE
4579
4580  015416  005367  163574          5$:     DEC     $TMP7           ;COUNT ONE CHAR READ OUT OF 64
4581  015422  003317                          BGT     3$              ;BR IF NOT LAST ONE
4582  015424  100546                          BMI     1$              ;;BR IF LAST ONE READ
4583  015426  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;DISABLE AUTO ECHO
4584  015434  000712                          BR      3$              ;GO READ LAST CHAR
4585
4586  015436  005167  163544          6$:     COM     $TMP3           ;TOGGLE I/O FLAG
4587  015442  001406                          BEQ     TST50           ;;BR IF DONE BOTH I/O DATA
4588  015444  005067  007474                  CLR     LINE            ;INIT LINE NO TO 00
4589  015450  012705  024556                  MOV     #RETAB0,R5      ;SET POINTER TO 0'S TABLE
4590  015454  000167  177452                  JMP     7$              ;REPEAT TEST FOR ZERO PATTERNS
4591
```

```
 4594                                   ;*****************************************************************
  (3)                                   ;*TEST 50       AUTO ECHO TEST 2 - ALL LINES
  (3)                                   ;*****************************************************************
  (2)  015460  000004                   TST50:  SCOPE
  (1)  015462  012767  000050  163544            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4595                                   .REM    %
  (1)                                   TEST ABSTRACT:
  (1)                                   **************
  (1)
  (1)                                           THIS TEST IS SIMILAR TO TEST 47 EXCEPT ALL SELECTED LINES OTHER
  (1)                                   THAN THE A.E. TEST LINE ARE ACTIVELY TURNING AROUND A BINARY COUNT
  (1)                                   TEST PATTERN IN NON-AUTO ECHO MODE AND THE A.E. TEST LINE IS TESTED
  (1)                                   FOR ALL 1'S DATA ONLY.
  (1)
  (1)                                   ERRORS:
  (1)                                   ********
  (1)
  (1)                                   1. [ERROR  32] IS CALLED TO REPORT A.E. TEST TIMEOUTS
  (1)                                   2. [ERROR  31] IS CALLED TO REPORT ALL DATA COMPARE ERRORS
  (1)
  (1)                                   SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
  (1)                                   *****
  (1)
  (1)                                   DEBUG:
  (1)                                   ******
  (1)
  (1)                                           REFER TO TEST 47
  (1)
  (1)                                   KEY LOGIC:
  (1)                                   **********
  (1)
  (1)                                           REFER TO TEST 47
  (1)
  (1)                                   %
 4596  015470  012767  015544  163412   2$:     MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 4597  015476  012705  024516                   MOV     #AETAB,R5       ;SET POINTER TO A.E. TEST DATA TABLE
 4598  015502  005067  006706                   CLR     LMSK1           ;INIT BIT TEST MASK
 4599  015506  000261                           SEC                     ;GENERATER MARKER BIT IN "C"
 4600  015510  000401                           BR      12$             ;GO SHIFT MASK
 4601  015512  000241                   1$:     CLC                     ;INIT THE "C" BIT
 4602  015514  006167  006674           12$:    ROL     LMSK1           ;SHIFT TEST BIT
 4603  015520  001410                           BEQ     11$             ;BR IF TESTED ALL LINES
 4604  015522  012567  163470                   MOV     (R5)+,$TMP7     ;GET THE A.E. TEST DATA FOR THIS LINE
 4605  015526  036767  006662  006654           BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
 4606  015534  001766                           BEQ     1$              ;BR IF NOT
 4607  015536  004767  005262                   JSR     PC,SELINE       ;GO SELECT A LINE
 4608  015542                           11$:
  (2)  015542  000571                           BR      TST51           ;;BR IF DONE ALL SELECTED LINES
 4609
 4610  015544  016701  006630           2$:     MOV     DHADR,R1        ;RESET DEVADR IN CASE OF ERROR LOOP
 4611  015550  012711  004000                   MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
 4612  015554  004767  006414                   JSR     PC,SETALL       ;GO SET UP FOR BINARY COUNT XFER ON
 4613                                                                   ;ALL LINES OTHER THAN THE SELECTED ONE
 4614  015560  156711  007360                   BISB    LINE,(R1)       ;SELECT THE LINE FOR A.E. TEST
```

```
4615  015564  010561  000006                  MOV     R5,CAR(R1)      ;SET BUS ADDR TO XMIT TEST CHAR
4616  015570  162761  000002  000006          SUB     #2,CAR(R1)      ;CORRECT THE ADDRESS
4617  015576  012761  177777  000010          MOV     #-1,BCR(R1)     ;XMIT ONE CHAR ON THIS LINE
4618  015604  012761  133503  000004          MOV     #133503,LPR(R1) ;DO IT AT 9600 BAUD/8 BITS
4619  015612  116767  163264  163364          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
4620  015620  042767  177400  163356          BIC     #177400,$TMP2
4621  015626  046767  006560  007022          BIC     LINMSK,LINACT   ;MAKE THIS LINE APPEAR INACTIVE
4622  015634  016761  006550  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTED TRANSMITTERS
4623
4624  015642  012767  000002  007322  21$:    MOV     #2,TIMEA        ;INIT TIMER A
4625  015650  005067  007320                  CLR     TIMEB           ;INIT TIMER B
4626  015654  016103  000002          3$:     MOV     NRC(R1),R3      ;  T     DATA
4627  015660  100414                  BMI     4$                      ;  IF
4628  015662  004767  006264                  JSR     PC,TIMEIT       ;  T   T  R
4629  015666  000772                  BR      3$                      ;TIMER ROUTINE WILL MOVE RETURN PC
4630                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4631
4632  015670  016167  000004  163302          MOV     LPR(R1),$TMP0   ;SAVE THE CURRENT "LPR"
4633  015676  004567  005214                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4634  015702  025144                          LINE
4635  015704  027607                          EM32+35
4636  015706  104032                  ERROR   32                      ;AUTO ECHO TIMEOUT
4637  015710  000700                  BR      1$                      ;GO TRY NEXT LINE
4638
4639  015712  010304          4$:     MOV     R3,R4                   ;EXTRACT LINE NUMBER OF RCVD CHAR
4640  015714  000304                  SWAB    R4
4641  015716  042704  177760                  BIC     #177760,R4
4642  015722  010402                  MOV     R4,R2                   ;SAVE IT IN R2
4643  015724  006302                  ASL     R2                      ;GENERATE TABLE INDEX IN R2
4644  015726  126704  007212                  CMPB    LINE,R4         ;IS THIS THE A.E. TEST LINE ??
4645  015732  001426                  BEQ     5$                      ;BR IF YES
4646
4647  015734  026203  032436                  CMP     RBUF(R2),R3     ;RECVD DATA CORRECT ??
4648  015740  001447                  BEQ     6$                      ;BR IF IT WAS
4649
4650  015742  004767  006366                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4651  015746  010467  163240                  MOV     R4,$TMP5        ;SAVE THE LINE NUMBER
4652  015752  016204  032436                  MOV     R  F(R2),R4     ;SET UP S/B DATA
4653  015756  062702  032436                  ADD     #  F,R2         ;SET UP S/B ADDRESS
4654  015762  012701  177703                  MOV     #177703,R1      ;SET UP THE WAS ADDRESS
4655  015766  004767  004704                  JSR     PC,SU  2R       ;GO SET UP ERROR INFO
4656  015772  004567  005120                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4657  015776  001212                          $TMP5
4658  016000  027452                          E 31+45
4659  016002  104031                  ERR R   31                      ;NON-ECHO DATA COMPARE ERROR
4660  016004  000167  177502                  JMP     1$              ;GO TRY NEXT LINE
4661
4662
4663  016010  020367  163202          5$:     CMP     R3,$TMP7        ;CHAR ECHOED OK ??
4664  016014  001427                  BEQ     7$                      ;BR IF YES
4665
4666  016016  004767  006312                  JSR     PC,SAPS         ;SA E THE ERROR PSW
4667  016022  012702  001216                  MOV     #$T  27,R2      ;SAVE THE S/B ADDRESS
4668  016026  016704  163164                  MOV     $TMP7,R4        ;SAVE THE S/B DATA
```

```
 4669  016032  012701  177703                      MOV     #177703,R1      ;SAVE THE WAS ADDRESS
 4670  016036  004767  004634                      JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4671  016042  004567  005050                      JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MESSAGE
 4672  016046  025144                              LINE
 4673  016050  027452                              EM31+45
 4674  016052  104031                              ERROR   31              ;AUTO ECHO LINE DATA ERROR
 4675  016054  000167  177432                      JMP     1$              ;GO TRY NEXT LINE
 4676
 4677  016060  105262  032436            6$:     INCB    RBUF(R2)        ;GENERATE NEXT EXPECTED DATA ON THIS LINE
 4678  016064  001366                              BNE     21$             ;BR IF ITS NOT BACK TO 000
 4679  016066  046267  024616  006562            BIC     LINBIT(R2),LINACT       ;INDICATE THIOS LINE DONE 256 BYTES
 4680  016074  005767  006556            7$:     TST     LINACT          ;ALL LINES INACTIVE
 4681  016100  001260                              BNE     21$             ;BR IF NOT
 4682  016102  042761  100000  000004            BIC     #BIT15,LPR(R1)  ;TURN OFF THE A.E. BIT
 4683  016110  105761  000017                      TSTB    SSR+1(R1)       ;SILO EMPTY ??
 4684  016114  001002                              BNE     8$              ;BR IF NOT
 4685  016116  000167  177370                      JMP     1$              ;GO TEST NEXT LINE
 4686  016122  000167  177514            8$:     JMP     21$             ;GO EMPTY IT
```

```
                                        ;*****************************************************************
                                        ;*TEST 51       AUTO ECHO TEST 3 - ALL LINES
                                        ;*****************************************************************
                                        TST51:  SCOPE
016126  000004
016130  012767  000051  163076                MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                        .REM    X
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST IS IDENTICAL TO TEST 47 EXCEPT ALL SELECTED LINES
                                        ARE ACTIVATED CONCURRENTLY RATHER THAN ONE AT A TIME AND ONLY
                                        THE ALL 1'S DATA IS USED.

                                        ERRORS:
                                        *******

                                        1. [ERROR  36] IS CALLED TO REPORT "DATA AVAIL" TIMEOUTS
                                        2. [ERROR  31] IS CALLED TO REPORT A.E. DATA ERRORS

                                        SYNC:   M7277   SHH LOAD BAR LB+HB L    CN2
                                        *****

                                        DEBUG:
                                        ******

                                                REFER TO TEST 47

                                        KEY LOGIC:
                                        **********

                                                REFER TO TEST 47

                                        %
016136  012767  016144  162744          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
016144  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
016150  012705  000020                  MOV     #20,R5          ;INIT COUNTER TO SET UP 16. LINES
016154  012702  024516                  MOV     #BETAB,R2       ;SET UP POINTER TO AUTO ECXHO TEST DATA
016160  012703  032376                  MOV     #RCNT,R3        ;R3 POINTS TO TABLE OF CHAR COUNTERS
016164  010261  000006          2$:     MOV     R2,CAR(R1)      ;SET UP BUS ADDRESS REG
016170  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET UP BYTE COUNT REG
016176  012761  131403  000004          MOV     #131403,LPR(R1) ;SET UP LINE PARAMETERS
016204  005023                  CLR     (R3)+           ;CLEAR A COUNTER
016206  062702  000002          ADD     #2,R2           ;UPDATE POINTERS
016212  005211                  INC     (R1)            ;SELECT NEXT LINE
016214  005305                  DEC     R5              ;COUNT ONE DONE
016216  001362                  BNE     2$              ;   TILL 16. DONE
016220  116767  162656  162756          MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
016226  042767  177400  162750          BIC     #177400,$TMP2
016234  016767  006150  006414          MOV     LINSEL,LINACT   ;SET FLAG TO INDICATE ALL 16. ACTIVE
016242  016761  006142  000012          MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL XMITTERS

016250  012767  000002  006714          MOV     #2,TIMEA        ;INIT TIMER A
016256  005067  006712                  CLR     TIMEB           ;INIT TIMERB
016262  105711                  3$:     TSTB    (R1)            ;CHAR AVAIL SET YET ?
```

```
4712  016264  100410                        BMI     4$              ;BR IF YES
4713  016266  004767  005660                JSR     PC,TIMEIT       ;CALL THE TIMER
4714  016272  000773                        BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4715                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4716
4717  016274  016167  000004  162676        MOV     LPR(R1),$TMP0   ;SAVE THE "LPR" REG
4718  016302  104036                        ERROR   36              ;DATA AVAILABLE TIMEOUT
4719  016304  000453                        BR      TST52           ;;EXIT TEST ON ERROR
4720
4721  016306  016103  000002          4$:   MOV     NRC(R1),R3      ;GET THE WAS DATA
4722  016312  010302                        MOV     R3,R2           ;BUILD AND SAVE LINE NO.
4723  016314  000302                        SWAB    R2
4724  016316  042702  177760                BIC     #177760,R2
4725  016322  010267  162666                MOV     R2,$TMP6        ;SAVE THE LINE NO.
4726  016326  006302                        ASL     R2              ;GENERATE TABLE OFFSET
4727  016330  005262  032376                INC     RCNT(R2)        ;COUNT THE CHARACTER
4728  016334  020362  024516                CMP     R3,AETAB(R2)    ;IS THE DATA CORRECT ??
4729  016340  001420                        BEQ     5$              ;BR IF YES
4730
4731  016342  004767  005766                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4732  016346  016204  024516                MOV     AETAB(R2),R4    ;GET THE S/B DATA
4733  016352  062702  024516                ADD     #AETAB,R2       ;GENERATE S/B ADDRESS
4734  016356  062701  000002                ADD     #NRC,R1         ;GENERATE THE WAS ADDRESS
4735  016362  004767  004310                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4736  016366  004567  004524                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4737  016372  001214                        $TMP6
4738  016374  027452                        EM31+45
4739  016376  104031                        ERROR   31              ;DATA COMPARE ERROR
4740  016400  000415                        BR      TST52           ;;EXIT TEST ON ERROR
4741
4742  016402  022762  000100  032376  5$:   CMP     #100,RCNT(R2)   ;DONE 64. CHARS ON THIS LINE ?
4743  016410  001324                        BNE     3$              ;BR IF NOT
4744  016412  016711  162576                MOV     $TMP6,(R1)      ;SELECT LINE IN SCR REG
4745  016416  042761  100000  000004        BIC     #BIT15,LPR(R1)  ;TURN OFF A.E. BIT
4746  016424  046267  024616  006224        BIC     LINBIT(R2),LINACT       ;ALL LINES INACTIVE ??
4747  016432  001313                        BNE     3$              ;BR IF NOT
```

```
 4750                                ;*****************************************************************
  (3)                                ;*TEST 52       BREAK BIT TEST - ALL LINES
  (3)                                ;*****************************************************************
  (2) 016434  000004                 TST52:  SCOPE
  (1) 016436  012767  000052  162570         MOV     #STN-1,STESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4751                                .REM    %
  (1)                                TEST ABSTRACT:
  (1)                                **************
  (1)
  (1)                                        THIS TEST VERIFIES THAT THE "BREAK" FEATURE WORKS PROPERLY
  (1)                                FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS FOLLOWS:
  (1)
  (1)                                        1. SET UP THE ERROR LOOP RETURN
  (1)                                        2. RETRIEVE THE CORRECT S/B DATA FROM THE "BREAK" DATA
  (1)                                           TABLE AND UPDATE THE POINTER.
  (1)                                        3. GO SELECT A LINE TO TEST - GO TO TEST 52 IF DONE ALL
  (1)                                           SELECTED LINES
  (1)                                        4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES.
  (1)                                        5. PRIME SELECTED LINE TO OUTPUT TWO "NULL" CHARS TO CLEAR UART
  (1)                                        6. ACTIVATE THE SELECTED LINE
  (1)                                        7. WAIT FOR SILO TO RECEIVE TWO NULLS - IF TIMEOUT REPORT ERROR
  (1)                                           AND RESTART AT STEP 2
  (1)                                        8. IF NO TIMEOUT CLEAR THE SELECTED DH11 AND RESELECT LINE NO.
  (1)                                        9. PRIME SELECTED LINE TO OUTPUT 256. CHARS.
  (1)                                        10.SET THE SELECTED LINE'S BREAK BIT
  (1)                                        11.ACTIVATE THE SELECTED LINE
  (1)                                        12.WAIT FOR "BCR" FLAG TO CLEAR -F TIMEOUT REPORT ERROR AND
  (1)                                           RESTART AT STEP 2
  (1)                                        13.IF NO TIMEOUT VERIFY THAT THE SILO RECEIVED ONLY ONE CHAR-
  (1)                                           IF NOT REPORT ERROR AND RESTART AT STEP 2
  (1)                                        14.IF SILO RECEIVED ONLY ONE CHAR VERIFY THAT IT WAS A "BREAK"
  (1)                                           CHAR - IF NOT REPORT ERROR - AND RESTART AT STEP 2
  (1)
  (1)                                ERRORS:
  (1)                                ********
  (1)
  (1)                                1. [ERROR  25] IS CALLED TO REPORT ALL ERRORS
  (1)
  (1)                                SYNC:   M7277   SH4     LOAD BCR H      FU1
  (1)                                *****
  (1)
  (1)                                DEBUG:
  (1)                                *******
  (1)
  (1)                                1. IF ALL LINES FAILED SUSPECT THAT THE M7277 IS NOT GENERATING THE
  (1)                                   BREAK CONTROL REG LOAD SIGNAL.
  (1)
  (1)                                2. IF ONLY ONE LINE FAILS SUSPECT THE BREAK CONTROL LOGIC ON THE M7278
  (1)
  (1)                                KEY SIGNALS:
  (1)                                ************
  (1)
  (1)                                        M7277   SH4     LOAD BCR H      FU1
  (1)
```

```
  (1)                                       M7278   SHS THRU SH8
  (1)
  (1)                                                 74175 REGISTER CHIPS E51, E38, E67, E60
  (1)                                                 7400 DRIVERS    E45, E46, E75, E76
  (1)
  (1)                                 X
4752  016444  012767  016524  162436          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4753  016452  012705  024720                  MOV     #BRKTAB,R5      ;SET UP POINTER TO BREAK DATA TABLE
4754  016456  005067  005732                  CLR     LMSK1           ;INIT BIT TEST MASK
4755  016462  000261                          SEC                     ;SET BIT MARKER IN "C"
4756  016464  000401                          BR      12$             ;GO SHIFT MASK
4757  016466  000241                  1$:     CLC                     ;INIT THE "C" BIT
4758  016470  006167  005720          12$:    ROL     LMSK1           ;SHIFT TEST MARKER
4759  016474  001411                          BEQ     11$             ;BR IF ALL LINES DONE
4760  016476  012504                          MOV     (R5)+,R4        ;GET TEST DATA FOR THIS LINE
4761  016500  036767  005710  005702          BIT     LMSK1,LINSEL    ;LINE SELECTED ?
4762  016506  001767                          BEQ     1$              ;BR IF NOT
4763  016510  004767  004310                  JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4764  016514  000401                          BR      11$             ;BR IF DONE ALL SELECTED LINES
4765  016516  000402                          BR      2$              ;GO TEST THE SELECTED LINE
4766  016520  000167  000454          11$:    JMP     9$              ;GO EXIT TEST
4767  016524  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
4768  016530  004767  004442                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
4769  016534  116711  006404                  MOVB    LINE,(R1)       ;SELECT THE LINE
4770
4771  016540  012761  025200  000006          MOV     #TNULL,CAR(R1)  ;SET UP TO OUTPUT TWO NULL CHARS
4772  016546  012761  177776  000010          MOV     #-2,BCR(R1)     ;SET BYTE COUNT TO 2
4773  016554  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
4774  016562  016761  005624  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
4775
4776  016570  012767  000001  006374          MOV     #1,TIMEA        ;INIT TIMER A
4777  016576  005067  006372                  CLR     TIMEB           ;INIT TIMER B
4778  016602  122761  000002  000017  3$:     CMPB    #2,SSR+1(R1)    ;TWO CHARS RECEIVED ??
4779  016610  001432                          BEQ     4$              ;BR IF YES
4780  016612  004767  005334                  JSR     PC,TIMEIT       ;CALL THE TIMER
4781  016616  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4782                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4783
4784  016620  004767  005510                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4785  016624  010467  162352                  MOV     R4,$TMP1        ;SAVE S/B DATA
4786  016630  010102                          MOV     R1,R2           ;SET UP REGADR
4787  016632  062702  000016                  ADD     #SSR,R2
4788  016636  011203                          MOV     (R2),R3         ;GET THE WAS DATA
4789  016640  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
4790  016644  012704  000002                  MOV     #2,R4           ;SET UP S/B DATA
4791  016650  000304                          SWAB    R4
4792  016652  004767  004020                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4793  016656  004567  004234                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
4794  016662  025144                          LINE
4795  016664  027155                          EM2S+34
4796  016666  016704  162310                  MOV     $TMP1,R4        ;RESTORE S/B DATA
4797  016672  104025                          ERROR   25              ;TIMED OUT WAITING FOR TWO NULLS
4798  016674  000674                          BR      1$              ;GO TRY NEXT LINE
4799
```

```
 4800  016676  012711  004000          4$:   MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
 4801  016702  116711  006236                MOVB    LINE,(R1)       ;SELECT THE LINE
 4802  016706  012761  033436  000006        MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG FOR XMITTR
 4803  016714  012761  177400  000010        MOV     #-400,BCR(R1)   ;SET BYTE COUNT TO XMIT 256(10) CHARS
 4804  016722  012761  033503  000004        MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
 4805  016730  016761  005456  000014        MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR ACTIVE LINE
 4806  016736  016761  005450  000012        MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4807
 4808  016744  012767  000005  006220        MOV     #5,TIMEA        ;INIT TIMER A
 4809  016752  005067  006216                CLR     TIMEB           ;INIT TIMER B
 4810  016756  005761  000012          5$:   TST     BAR(R1)         ;BAR BIT CLEARED ??
 4811  016762  001426                BEQ     6$              ;BR IFO YES
 4812  016764  004767  005162                JSR     PC,TIMEIT       ;CALL THE TIMER
 4813  016770  000772                BR      5$              ;TIMER ROUTINE WILL MOVE RETURN PC
 4814                                                        ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4815
 4816  016772  004767  005336                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4817  016776  010467  162200                MOV     R4,$TMP1        ;SAVE THE S/B DATA
 4818  017002  010102                MOV     R1,R2           ;SET UP REGADR
 4819  017004  062702  000012                ADD     #BAR,R2
 4820  017010  011203                MOV     (R2),R3         ;GET THE WAS DATA
 4821  017012  005004                CLR     R4              ;SET UP S/B DATA
 4822  017014  004767  003656                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4823  017020  004567  004072                JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
 4824  017024  025144                LINE
 4825  017026  027155                EM25+34
 4826  017030  016704  162146                MOV     $TMP1,R4        ;RESTORE THE S/B DATA
 4827  017034  104025                ERROR   25              ;BAR BIT FAILED TO CLEAR
 4828  017036  000613                BR      1$              ;GO TRY NEXT LINE
 4829
 4830  017040  122761  000001  000017  6$:   CMPB    #1,SSR+1(R1)    ;ONE CHAR RECEIVED ?
 4831  017046  001430                BEQ     7$              ;BR IF YES
 4832
 4833  017050  004767  005260                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4834  017054  010467  162122                MOV     R4,$TMP1        ;SAVE THE S/B DATA
 4835  017060  010102                MOV     R1,R2           ;SET UP REGADR
 4836  017062  062702  000016                ADD     #SSR,R2
 4837  017066  011203                MOV     (R2),R3         ;GET THE WAS DATA
 4838  017070  042703  100377                BIC     #100377,R3      ;CLEAR JUNK
 4839  017074  012704  000001                MOV     #1,R4           ;SET UP S/B DATA
 4840  017100  000304                SWAB    R4
 4841  017102  004767  003570                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4842  017106  004567  004004                JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
 4843  017112  025144                LINE
 4844  017114  027155                EM25+34
 4845  017116  016704  162060                MOV     $TMP1,R4        ;RESTORE THE S/B DATA
 4846  017122  104025                ERROR   25              ;FAILED TO RECEIVE THE ONE CHAR
 4847  017124  000167  177336                JMP     1$              ;GO TRY NEXT LINE
 4848
 4849
 4850  017130  016103  000002          7$:   MOV     RBC(R1),R3      ;GET THE WAS DATA
 4851  017134  020304                CMP     R3,R4           ;WAS IT A BREAK CHAR ?
 4852  017136  001002                BNE     8$              ;BR IF NOT CORRECT
 4853  017140  000167  177322                JMP     1$              ;GO TEST NEXT LINE
```

```
4854
4855  017144  004767  005164          8$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
4856  017150  010102                          MOV     R1,R2           ;SET UP REGADR
4857  017152  062702  000002                  ADD     #NRC,R2
4858  017156  004767  003514                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4859  017162  004567  003730                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
4860  017166  025144                          LINE
4861  017170  027155                          EM25+34
4862  017172  104025                          ERROR   25              ;INCORRECT DATA RECEIVED
4863  017174  000167  177266                  JMP     1$              ;GO TRY NEXT LINE
4864  017200  000240                  9$:     NOP                     ;EXIT THIS TEST
4865
```

```
                                ;*****************************************************************
                                ;*TEST 53       HALF DUPLEX TEST - ALL LINES
                                ;*****************************************************************
017202  000004                  TST53:  SCOPE
017204  012767  000053  162022          MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    X
                                TEST ABSTRACT:
                                *************

                                        THIS TEST VERIFIES THAT THE RECEIVERS ON ALL SELECTED LINES ARE
                                "BLINDED" WHEN THE HALF-DUPLEX MODE IS ENABLED. THE TEST
                                SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. GO SELECT A LINE NO. TO TEST
                                        3. IF DONE ALL SELECTED LINES - GO TO TEST 53
                                        4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES
                                        5. PRIME THE SELECTED DH11 TO XMIT 256. CHARS IN HALF-
                                           DUPLEX MODE.
                                        6. ACTIVATE THE SELECTED LINE AND WAIT FOR THE "BAR" REG TO CLEAR
                                        7. IF TIMEOUT - REPORT ERROR AND GO TO STEP 2
                                        8. IF NO TIMEOUT VERIFY THE "CHAR AVAIL" DID NOT SET (RECEIVER
                                           BLINDED) - IF ERROR REPORT IT AND GO TO STEP2 - IF NO ERROR
                                            GO TO STEP 2

                                ERRORS:
                                *******

                                1. [ERROR  26] IS CALLED TO REPORT ALL ERRORS

                                SYNC:   M7277   SH3     INIT A H        EF2
                                *****

                                DEBUG:
                                ******

                                1. SUSPECT EITHER THE M7289 OR THE M7288 MODULES

                                KEY LOGIC:
                                **********

                                        M7289   SH5     HALF DUPLEX <15:00> H SIGNALS
                                                        END OF CHAR <15:00> SIGNALS

                                        M7288   SH5     HALF DUPLEX <03:00> H SIGNALS
                                                SH7     "       "   <07:00> H SIGNALS
                                                SH9     "       "   <11:08> H SIGNALS
                                                SH11    "       "   <15:12> H SIGNALS

                                %
017212  012767  017226  161670          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
017220  004767  003600          1$:     JSR     PC,$ELINE
017224  000477                  2$:     BR      TST54           ;;BR IF ALL LINES TESTED
017226  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
```

```
017232  004767  003740              JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
017236  156711  005702              BISB    LINE,(R1)       ;SELECT THE LINE
017242  012761  033436  000006      MOV     #TBUF,CAR(R1)   ;POINT TO XMIT BUFFER
017250  012761  177400  000010      MOV     #-400,BCR(R1)   ;XMIT 256(10) CHARS
017256  012761  073503  000004      MOV     #73503,LPR(R1)  ;SET UP THE LINE PARAMETERS
017264  016761  005122  000012      MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

017272  012767  000001  005672      MOV     #1,TIMEA        ;INIT TIME A
017300  005067  005670              CLR     TIMEB           ;INIT TIME B
017304  005761  000012      3$:     TST     BAR(R1)         ;WAIT FOR XMITTR TO FINISH
017310  001423                      BEQ     4$              ;BR IF XMITTR FINISHED
017312  004767  004634              JSR     PC,TIMEIT       ;CALL TIMER
017316  000772                      BR      3$              ;TIMER WILL MOVE RETURN PC AROUND
                                                            ;THIS BRANCH IF TIMEOUT OCCURS

017320  004767  005010              JSR     PC,SAPS         ;SAVE THE ERROR PSW
017324  016103  000012              MOV     BAR(R1),R3      ;GET THE WAS DATA
017330  010102                      MOV     R1,R2           ;SET UP REGADR
017332  062702  000012              ADD     #BAR,R2
017336  005004                      CLR     R4              ;SET UP NEW S/B DATA
017340  004767  003332              JSR     PC,SUER2A       ;GO SET UP THE ERROR INFO
017344  004567  003546              JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
017350  025144                      LINE
017352  027220                      EM26+37
017354  104026                      ERROR   26              ;BAR BIT FAILED TO CLEAR ON TIME
017356  000720                      BR      1$              ;GO TRY NEXT LINE

017360  105711          4$:         TSTB    (R1)            ;CHAR AVAIL SET ??
017362  100316                      BPL     1$              ;BR IF NOT IT SHOULDN'T BE

017364  004767  004744              JSR     PC,SAPS         ;SAVE THE ERROR PSW
017370  010102                      MOV     R1,R2           ;SET UP REGADR
017372  011103                      MOV     (R1),R3         ;GET WAS DATA
017374  042703  100000              BIC     #BIT15,R3       ;CLEAR JUNK BIT
017400  116704  005540              MOVB    LINE,R4         ;SET UP S/B DATA
017404  004767  003266              JSR     PC,SUER2A       ;GO SETUP ERROR INFO
017410  004567  003502              JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
017414  025144                      LINE
017416  027220                      EM26+37
017420  104026                      ERROR   26              ;HALF DUPLEX FAILED TO BLIND RECVR
017422  000676                      BR      1$              ;GO SELECT NEXT LINE
```

```
 4917                                   ;*****************************************************************
  (3)                                   ;*TEST 54       VERIFY THAT OVERRUN CAN SET PROPERLY - ALL LINES
  (3)                                   ;*****************************************************************
  (2)  017424  000004                   TST54:  SCOPE
  (1)  017426  012767  000054  161600           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4918                                   .REM    X
  (1)                                   TEST ABSTRACT:
  (1)                                   ***************
  (1)
  (1)                                           THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
  (1)                                   THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
  (1)                                   PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:
  (1)
  (1)                                           1. SET UP THE ERROR LOOP RETURN
  (1)                                           2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
  (1)                                              END OF PASS HANDLER.
  (1)                                           3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
  (1)                                           4. ACTIVATE THE SELECTED LINE
  (1)                                           5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
  (1)                                              AND RESTART AT STEP 2
  (1)                                           6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
  (1)                                              "OVERRUN" IS SET ON THE LAST WORD READ
  (1)                                           7. IF NOT REPORT ERROR AND RESTART AT STEP 2
  (1)
  (1)                                   ERRORS:
  (1)                                   ********
  (1)
  (1)                                   1. [ERROR  50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
  (1)                                   2. [ERROR  56] IS CALLEDT TO REPORT "OVERRUN" ERROR
  (1)
  (1)                                   SYNC:   M7277   SH3     INIT A H        EF2
  (1)                                   *****
  (1)
  (1)                                   DEBUG:
  (1)                                   ******
  (1)
  (1)                                   1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
  (1)                                      FOR THE APPROPRIATE LINE IN QUESTION.
  (1)
  (1)                                   2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE
  (1)
  (1)                                   KEY LOGIC:
  (1)                                   **********
  (1)
  (1)                                           M7279   SH1     MASTER OR H     E12-9
  (1)                                                   SH2     MEMORY CHIP (3341) E13-11
  (1)
  (1)                                           M7280   SH2     UC1 OR 2 MASTER OR      EN2
  (1)                                                   SH2-5   UART PIN 15 (BUF OR LINE NN)
  (1)
  (1)                                   %
  (1)
 4919  017434  012767  017450  161446           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 4920  017442  004767  003356          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE # TO TEST
```

```
 4921  017446  000512                          BR      ENDA            ;;BR IF DONE ALL SELECTED LINES
 4922  017450  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DHII
 4923  017454  116711  005464                  MOVB    LINE,(R1)       ;SELECT THE LINE TO TEST
 4924  017460  012761  033436  000006          MOV     #TBUF,CAR(R1)   ;SET UP CURRENT ADDRESS
 4925  017466  012761  177674  000010          MOV     #-68.,BCR(R1)   ;SET UP BYTE COUNT REG
 4926  017474  012761  033503  000004          MOV     #33503,LPR(R1)  ;DO IT AT 9600 BAUD - 8 BITS
 4927  017502  016761  004704  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4928
 4929  017510  012767  000001  005454          MOV     #1,TIMEA        ;INIT TIMERS A AND B
 4930  017516  005067  005452                  CLR     TIMEB
 4931  017522  005711                  3$:     TST     (R1)            ;TRANSMITTER DONE ??
 4932  017524  100425                          BMI     4$              ;BR IF YES
 4933  017526  004767  004420                  JSR     PC,TIMEIT       ;CALL TIMER
 4934  017532  000773                          BR      3$              ;BR IF NO TIMEOUT
 4935
 4936  017534  004767  004574                  JSR     PC,SAPS         ;GO SAVE PSW
 4937  017540  011103                          MOV     (R1),R3         ;GET THE WAS DATA
 4938  017542  042703  077760                  BIC     #77760,R3       ;CLEAR UNINTERESTING BITS
 4939  017546  116704  005372                  MOVB    LINE,R4         ;SET UP S/B DATA
 4940  017552  052704  100000                  BIS     #BIT15,R4
 4941  017556  010102                          MOV     R1,R2           ;SET UP REGADR
 4942  017560  004767  003112                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4943  017564  004567  003326                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE HEADER
 4944  017570  025144                          LINE
 4945  017572  031076                          EM50+53
 4946  017574  104050                          ERROR   50              ;REPORT XMIT DONE TIME OUT
 4947  017576  000721                          BR      1$              ;GO TRY NEXT LINE
 4948
 4949  017600  012767  000101  161374  4$:     MOV     #65.,$TMP1      ;SET UP TO READ 65. WORDS FROM SILO
 4950  017606  116704  005332                  MOVB    LINE,R4         ;SET UP S/B DATA
 4951  017612  000304                          SWAB    R4
 4952  017614  152704  000101                  BISB    #65.,R4
 4953  017620  052704  140000                  BIS     #BIT15+BIT14,R4 ;PUT IN OVERRUN AND VALID DATA BITS
 4954  017624  016103  000002          5$:     MOV     NRC(R1),R3      ;GET WAS DATA FROM SILO
 4955  017630  005367  161346                  DEC     $TMP1           ;COUNT ONE WORD READ
 4956  017634  001373                          BNE     5$              ;BR TIL 65. READ
 4957  017636  020304                          CMP     R3,R4           ;WAS DATA AND OVERRUN CORRECT ??
 4958  017640  001700                          BEQ     1$              ;BR IF YES TRY NEXT SELECTED LINE
 4959
 4960  017642  004767  004466                  JSR     PC,SAPS         ;GO SAVE PSW
 4961  017646  010102                          MOV     R1,R2           ;SET UP REGADR
 4962  017650  062702  000002                  ADD     #NRC,R2
 4963  017654  004767  003016                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4964  017660  004567  003232                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG HDR
 4965  017664  025144                          LINE
 4966  017666  031514                          EM56+42
 4967  017670  104056                          ERROR   56              ;OVERRUN OR DATA INCORRECT
 4968  017672  000663                          BR      1$              ;GO TEST NEXT SELECTED LINE
 4969
```

```
 4972  017674  000004                        ENDA:   SCOPE
 4973  017676  012767  000240  000054                MOV     #240,SEOP       ;NOP THE SCOPE AT THE BEGINNING OF EOP
 4974  017704  005267  005232                        INC     DHNUM           ;GENERATE NEW DH11 NUMBER
 4975  017710  062767  000002  005232                ADD     #2,AOPTR        ;UPDATE THE TABLE POINTERS
 4976  017716  062767  000002  005226                ADD     #2,VCPTR
 4977  017724  062767  000002  005222                ADD     #2,BRPTR
 4978  017732  006367  004446                        ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
 4979  017736  001410                                BEQ     $EOP            ;BR IF TESTED ALL SELECTED DH11'S
 4980  017740  036767  004440  004440                BIT     SELMSK,DHSEL    ;IS THIS DH11 SELECTED ?
 4981  017746  001752                                BEQ     ENDA            ;BR IF NOT
 4982  017750  105067  161126                        CLRB    $TSTNM          ;INIT TEST NUMBER
 4983  017754  000167  162632                        JMP     RSTRTA          ;GO TEST THIS DH11
 4984                                  ;******************************************************************
  (1)
  (1)                                  .SBTTL  END OF PASS ROUTINE
  (1)
  (1)                                  ;*INCREMENT THE PASS NUMBER ($PASS)
  (1)                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
  (1)                                  ;*IF THERES A MONITOR GO TO IT
  (1)                                  ;*IF THERE ISN'T JUMP TO START2
  (1)
  (1)  017760                          SEOP:
  (1)  017760  000004                          SCOPE
  (1)  017762  005067  161114                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
  (1)  017766  005067  161226                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (1)  017772  005267  161240                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
  (1)  017776  042767  100000  161232          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
  (1)  020004  005327                          DEC     (PC)+           ;;LOOP?
  (1)  020006  000001                  SEOPCT: .WORD   1
  (1)  020010  003022                          BGT     $DOAGN          ;;YES
  (1)  020012  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (1)  020014  000001                  SENDCT: .WORD   1
  (1)  020016  020006                          $EOPCT
  (1)  020020  104400  020062                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
  (2)  020024  016746  161206                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
  (2)  020030  104404                          TYPOS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
  (1)  020032  104400  020077                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
  (1)  020036                          $GET42:
  (1)
  (1)  020036  013700  000042                  MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (1)  020042  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  (1)  020044  000005                          RESET                   ;;CLEAR THE WORLD
  (1)  020046  004710                  SENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (1)  020050  000240                          NOP                     ;;SAVE ROOM
  (1)  020052  000240                          NOP                     ;;FOR
  (1)  020054  000240                          NOP                     ;ACT11
  (1)  020056                          $DOAGN:
  (1)  020056  000137  002506                  JMP     @#START2                ;;RETURN
  (1)  020062  005015  047105  020104  SENDMG: .ASCIZ  <15><12>/END PASS #/
  (1)  020070  040520  051523  021440
  (1)  020076     000
  (1)  020077     377     377     000  SENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
 4985                                  ;******************************************************************
  (1)
```

```
(1)                                     .SBTTL  SCOPE HANDLER ROUTINE
(1)
(1)                                     ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)                                     ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1)                                     ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1)                                     ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                     ;*SW14=1        LOOP ON TEST
(1)                                     ;*SW11=1        INHIBIT ITERATIONS
(1)                                     ;*SW09=1        LOOP ON ERROR
(1)                                     ;*SW08=1        LOOP ON TEST IN SWR<7:0>
(1)                                     ;*CALL
(1)                                     ;*      SCOPE           ;;SCOPE=IOT
(1)
(1)  020102                             $SCOPE:
(3)  020102  005067  005036             1$:     CLR     LINE            ;INIT THE LINE NO. TO ZERO
(3)  020106  016701  004266                     MOV     DVAR,R1         ;SET UP DEVAR IN R1
(1)  020112  032777  040000  161016             BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
(1)  020120  001114                             BNE     $OVER           ;;YES IF SW14=1
(1)                                     ;#####START OF CODE FOR THE XOR TESTER#####
(1)  020122  000416                     $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(1)                                                                     ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(1)  020124  013746  000004                     MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)  020130  012737  020150  000004             MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(1)  020136  005737  177060                     TST     @#177060        ;;TIME OUT ON XOR?
(1)  020142  012637  000004                     MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  020146  000463                             BR      $SVLAD          ;;GO TO THE NEXT TEST
(1)  020150  022626                     5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(1)  020152  012637  000004                     MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  020156  000423                             BR      7$              ;;LOOP ON THE PRESENT TEST
(1)  020160                             6$:;#####END OF CODE FOR THE XOR TESTER#####
(1)  020160  032777  000400  160750             BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
(1)  020166  001404                             BEQ     2$              ;;BR IF NO
(1)  020170  127767  160742  160704             CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
(1)  020176  001465                             BEQ     $OVER           ;;BR IF YES
(1)  020200  105767  160677             2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(1)  020204  001421                             BEQ     3$              ;;BR IF NO
(1)  020206  126767  160703  160667             CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
(1)  020214  101015                             BHI     3$              ;;BR IF NO
(1)  020216  032777  001000  160712             BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
(1)  020224  001404                             BEQ     4$              ;;BR IF NO
(1)  020226  016767  160656  160652  7$:        MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
(1)  020234  000446                             BR      $OVER
(1)  020236  105067  160641             4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(1)  020242  005067  160752                     CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1)  020246  000415                             BR      1$              ;;ESCAPE TO THE NEXT TEST
(1)  020250  032777  004000  160660  3$:        BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
(1)  020256  001011                             BNE     1$              ;;BR IF YES
(1)  020260  005767  160752                     TST     $PASS           ;;IF FIRST PASS OF PROGRAM
(1)  020264  001 006                            BEQ     1$              ;;      INHIBIT ITERATIONS
(1)  020266  005 267  160612                    INC     $ICNT           ;;INCREMENT ITERATION COUNT
(1)  020272  020 767  160722  160604            CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
(1)  020300  002024                             BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
(1)  020302  012767  000001  160574  1$:        MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
(1)  020310  016767  000052  160702             MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
```

```
(1)  020316  105267  160560            $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
(1)  020322  116767  160554  160704            MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
(1)  020330  011667  160552                     MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
(1)  020334  011667  160550                     MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  020340  005067  160656                     CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  020344  112767  000001  160543            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  020352  016777  160524  160560  $OVER:    MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  020360  016716  160522                     MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  020364  000002                             RTI                     ;;FIXES PS
(1)  020366  000010                     $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
4986                                    ;*******************************************************************
(1)
(1)                                     .SBTTL  ERROR HANDLER ROUTINE
(1)
(1)                                     ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1)                                     ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1)                                     ;*AND GO TO $ERRTYP ON ERROR
(1)                                     ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                     ;*SW15=1        HALT ON ERROR
(1)                                     ;*SW13=1        INHIBIT ERROR TYPEOUTS
(1)                                     ;*SW09=1        LOOP ON ERROR
(1)                                     ;*CALL
(1)                                     ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1)
(1)  020370                             $ERROR:
(1)  020370  105267  160507             7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
(1)  020374  001775                             BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
(1)  020376  016777  160500  160534             MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1)  020404  005267  160502                     INC     $ERTTL          ;;INC THE ERROR COUNT
(1)  020410  011667  160502                     MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
(1)  020414  162767  000002  160474             SUB     #2,$ERRPC
(1)  020422  117767  160470  160464             MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
(1)  020430  032777  020000  160500             BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
(1)  020436  001004                             BNE     20$             ;;SKIP TYPEOUTS
(1)  020440  004767  000072                     JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
(1)  020444  104400  001225                     TYPE    ,$CRLF
(1)  020450                             20$:
(1)  020450  122767  000001  160572             CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  020456  001007                             BNE     2$              ;;NO,SKIP APT ERROR REPORT
(1)  020460  116767  160430  000004             MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
(1)  020466  004767  001146                     JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
(1)  020472     000             21$:    .BYTE   0
(1)  020473     000                     .BYTE   0
(1)  020474  000777             22$:    BR      22$             ;;APT ERROR LOOP
(1)  020476  005777  160434             2$:     TST     @SWR            ;;HALT ON ERROR
(1)  020502  100001                             BPL     3$              ;;SKIP IF CONTINUE
(1)  020504  000000                             HALT                    ;;HALT ON ERROR!
(1)  020506  032777  001000  160422     3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
(1)  020514  001402                             BEQ     4$              ;;BR IF NO
(1)  020516  016716  160366                     MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
(1)  020522  005767  160474             4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
(1)  020526  001402                             BEQ     5$              ;;BR IF NONE
(1)  020530  016716  160466                     MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1)  020534                             5$:
```

```
  (1)  020534  000002                        RTI                             ;;RETURN
 4987
  (1)                                ;*********************************************************************
  (1)
  (1)                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
  (1)
  (1)                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
  (1)                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
  (1)                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
  (1)
  (1)  020536                        $ERRTYP:
  (1)  020536  104400  001225                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020542  010046                        MOV     R0,-(SP)                ;;SAVE R0
  (1)  020544  005000                        CLR     R0                      ;;PICKUP THE ITEM INDEX
  (1)  020546  153700  001114                BISB    @#$ITEMB,R0
  (1)  020552  001004                        BNE     1$                      ;;IF ITEM NUMBER IS ZERO, JUST
  (1)                                                                        ;;TYPE THE PC OF THE ERROR
  (2)  020554  016746  160336                MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
  (2)                                                                        ;;ERROR ADDRESS
  (2)  020560  104401                        TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  020562  000426                        BR      6$                      ;;GET OUT
  (1)  020564  005300                1$:     DEC     R0                      ;;ADJUST THE INDEX SO THAT IT WILL
  (1)  020566  006300                        ASL     R0                      ;;      WORK FOR THE ERROR TABLE
  (1)  020570  006300                        ASL     R0
  (1)  020572  006300                        ASL     R0
  (1)  020574  062700  001354                ADD     #$ERRTB,R0              ;;FORM TABLE POINTER
  (1)  020600  012067  000004                MOV     (R0)+,2$                ;;PICKUP "ERROR MESSAGE" POINTER
  (1)  020604  001404                        BEQ     3$                      ;;SKIP TYPEOUT IF NO POINTER
  (1)  020606  104400                        TYPE                            ;;TYPE THE "ERROR MESSAGE"
  (1)  020610  000000                2$:     .WORD   0                       ;;"ERROR MESSAGE" POINTER GOES HERE
  (1)  020612  104400  001225                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020616  012067  000004        3$:     MOV     (R0)+,4$                ;;PICKUP "DATA HEADER" POINTER
  (1)  020622  001404                        BEQ     5$                      ;;SKIP TYPEOUT IF 0
  (1)  020624  104400                        TYPE                            ;;TYPE THE "DATA HEADER"
  (1)  020626  000000                4$:     .WORD   0                       ;;"DATA HEADER" POINTER GOES HERE
  (1)  020630  104400  001225                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020634  011000                5$:     MOV     (R0),R0                 ;;PICKUP "DATA TABLE" POINTER
  (1)  020636  001004                        BNE     7$                      ;;GO TYPE THE DATA
  (1)  020640  012600                6$:     MOV     (SP)+,R0                ;;RESTORE R0
  (1)  020642  104400  001225                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020646  000207                        RTS     PC                      ;;RETURN
  (2)  020650                        7$:
  (2)  020650  013046                        MOV     @(R0)+,-(SP)            ;;SAVE @(R0)+ FOR TYPEOUT
  (2)  020652  104401                        TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  020654  005710                        TST     (R0)                    ;;IS THERE ANOTHER NUMBER?
  (1)  020656  001770                        BEQ     6$                      ;;BR IF NO
  (1)  020660  104400  020666                TYPE    ,8$                     ;;TYPE TWO(2) SPACES
  (1)  020664  000771                        BR      7$                      ;;LOOP
  (1)  020666  020040      000       8$:     .ASCIZ  /  /                    ;;TWO(2) SPACES
  (1)          020672                        .EVEN
 4988                                ;*********************************************************************
  (1)
  (1)                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
  (1)
  (1)                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
```

```
(1)                                     ;*OCTAL (ASCII) NUMBER AND TYPE IT.
(1)                                     ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOS                   ;;CALL FOR TYPEOUT
(1)                                     ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1)                                     ;*      .BYTE   M               ;;M=1 OR 0
(1)                                     ;*                                      ;;1=TYPE LEADING ZEROS
(1)                                     ;*                                      ;;0=SUPPRESS LEADING ZEROS
(1)                                     ;*
(1)                                     ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)                                     ;*$TYPOS OR $TYPOC
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPON                   ;;CALL FOR TYPEOUT
(1)                                     ;*
(1)                                     ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOC                   ;;CALL FOR TYPEOUT
(1)
(1)  020672  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
(1)  020676  116667  000001  000211            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
(1)  020704  112667  000207                     MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
(1)  020710  062716  000002                     ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
(1)  020714  000406                             BR      $TYPON
(1)  020716  112767  000001  000171  $TYPOC:   MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
(1)  020724  112767  000006  000165            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
(1)  020732  112767  000005  000154  $TYPON:   MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
(1)  020740  010346                             MOV     R3,-(SP)        ;;SAVE R3
(1)  020742  010446                             MOV     R4,-(SP)        ;;SAVE R4
(1)  020744  010546                             MOV     R5,-(SP)        ;;SAVE R5
(1)  020746  116704  000145                     MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
(1)  020752  005404                             NEG     R4
(1)  020754  062704  000006                     ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
(1)  020760  110467  000132                     MOVB    R4,$OMODE       ;;SAVE IT FOR USE
(1)  020764  116704  000125                     MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
(1)  020770  016605  000012                     MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
(1)  020774  005003                             CLR     R3              ;;CLEAR THE OUTPUT WORD
(1)  020776  006105                  1$:        ROL     R5              ;;ROTATE MSB INTO "C"
(1)  021000  000404                             BR      3$              ;;GO DO MSB
(1)  021002  006105                  2$:        ROL     R5              ;;FORM THIS DIGIT
(1)  021004  006105                             ROL     R5
(1)  021006  006105                             ROL     R5
(1)  021010  010503                             MOV     R5,R3
(1)  021012  006103                  3$:        ROL     R3              ;;GET LSB OF THIS DIGIT
(1)  021014  105367  000076                     DECB    $OMODE          ;;TYPE THIS DIGIT?
(1)  021020  100016                             BPL     7$              ;;BR IF NO
(1)  021022  042703  177770                     BIC     #177770,R3      ;;GET RID OF JUNK
(1)  021026  001002                             BNE     4$              ;;TEST FOR 0
(1)  021030  005704                             TST     R4              ;;SUPPRESS THIS 0?
(1)  021032  001403                             BEQ     5$              ;;BR IF YES
(1)  021034  005204                  4$:        INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  021036  052703  000060                     BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
```

```
(1)  021042  052703  000040          5$:     BIS     #'  ,R3         ;;MAKE ASCII IF NOT ALREADY
(1)  021046  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  021052  104400  021112                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  021056  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
(1)  021062  003347                          BGT     2$              ;;BR IF MORE TO DO
(1)  021064  002402                          BLT     6$              ;;BR IF DONE
(1)  021066  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  021070  000744                          BR      2$              ;;GO DO THE LAST DIGIT
(1)  021072  012615                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)  021074  012604                          MOV     (SP)+,R4        ;;RESTORE R4
(1)  021076  012603                          MOV     (SP)+,R3        ;;RESTORE R3
(1)  021100  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  021106  012616                          MOV     (SP)+,(SP)
(1)  021110  000002                          RTI                     ;;RETURN
(1)  021112     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  021113     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  021114     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  021115     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)  021116  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
4989                                 ;*****************************************************************
(1)                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(1)                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1)                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(1)                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(1)                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(1)                                  ;*REPLACED WITH SPACES.
(1)                                  ;*CALL:
(1)                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
(1)                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
(1)
(2)  021120                          $TYPDS:
(3)  021120  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  021122  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  021124  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  021126  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  021130  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  021132  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
(1)  021136  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
(1)  021142  100004                          BPL     1$              ;;BR IF INPUT IS POS.
(1)  021144  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
(1)  021146  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
(1)  021154  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
(1)  021156  012703  021334                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
(1)  021162  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
(1)  021166  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
(1)  021170  016001  021324                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
(1)  021174  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
(1)  021176  002402                          BLT     4$              ;;BR IF DONE
(1)  021200  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
(1)  021202  000774                          BR      3$
(1)  021204  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
(1)  021206  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
```

```
(1)  021210  001002                          BNE     5$              ;;FALL THROUGH IF 0
(1)  021212  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
(1)  021214  100407                          BMI     7$              ;;BR IF YES
(1)  021216  106316                  5$:     ASLB    (SP)            ;;MSD?
(1)  021220  103003                          BCC     6$              ;;BR IF NO
(1)  021222  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
(1)  021230  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
(1)  021234  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)  021240  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)  021242  005720                          TST     (R0)+           ;;JUST INCREMENTING
(1)  021244  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
(1)  021250  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
(1)  021252  003002                          BGT     8$              ;;GO TO EXIT
(1)  021254  010502                          MOV     R5,R2           ;;GET THE LSD
(1)  021256  000764                          BR      6$              ;;GO CHANGE TO ASCII
(1)  021260  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
(1)  021262  100003                          BPL     9$              ;;BR IF NO
(1)  021264  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
(1)  021272  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
(3)  021274  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  021276  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  021300  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  021302  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  021304  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  021306  104400  021334                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  021312  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  021320  012616                          MOV     (SP)+,(SP)
(1)  021322  000002                          RTI                     ;;RETURN TO USER
(1)  021324  023420                  $DTBL:  10000.
(1)  021326  001750                          1000.
(1)  021330  000144                          100.
(1)  021332  000012                          10.
(1)  021334  000004                  $DBLK:  .BLKW   4
4990                                 ;*****************************************************************
(1)                                  
(1)                                  .SBTTL  TYPE ROUTINE
(1)                                  
(1)                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                  ;*
(1)                                  ;*CALL:
(1)                                  ;*1) USING A TRAP INSTRUCTION
(1)                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                  ;*OR
(1)                                  ;*      TYPE
(1)                                  ;*      MESADR
(1)                                  ;*
(1)
(1)  021344  105767  157605          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  021350  100002                          BPL     1$              ;;BR IF YES
(1)  021352  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
```

```
(1)  021354  000430                        BR      3$              ;;LEAVE
(1)  021356  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  021360  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
(1)  021364  122767  000001  157656        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  021372  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
(1)  021374  132767  000100  157647        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
(1)  021402  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
(1)  021404  010067  000004                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
(1)  021410  004767  000214                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
(1)  021414  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
(1)  021416  132767  000040  157625  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
(1)  021424  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
(1)  021426  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  021430  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  021432  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  021434  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
(1)  021436  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  021442  000002                        RTI                     ;;RETURN
(1)  021444  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
(1)  021450  001426                        BEQ     8$
(1)  021452  122716  000200                CMPB    #TCRLF,(SP)     ;;BRANCH IF NOT <CRLF>
(1)  021456  001004                        BNE     5$
(1)  021460  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
(1)  021462  104400                        TYPE                    ;;TYPE A CR AND LF
(1)  021464  001225                        $CRLF
(1)  021466  000757                        BR      2$              ;;GET NEXT CHARACTER
(1)  021470  004767  000056        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1)  021474  126726  157454        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1)  021500  001352                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1)  021502  016746  157444                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                                 ;;AND THE NULL CHAR.
(1)  021506  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1)  021512  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  021514  004767  000032                JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1)  021520  105367  000072                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  021524  000770                        BR      7$              ;;LOOP
(1)
(1)                                ;HORIZONTAL TAB PROCESSOR
(1)
(1)  021526  112716  000040        8$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
(1)  021532  004767  000014        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  021536  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  021544  001372                        BNE     9$              ;;TAB STOP
(1)  021546  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
(1)  021550  000726                        BR      2$              ;;GET NEXT CHARACTER
(1)  021552  105777  157370        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  021556  100375                        BPL     $TYPEC
(1)  021560  116677  000002  157362        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  021566  122766  000015  000002        CMPB    #15,2(SP)       ;;BRANCH IF
(1)  021574  001003                        BNE     1$              ;;NOT <CR>
(1)  021576  105067  000014                CLRB    $CHARCNT        ;;
(1)  021602  000406                        BR      $TYPEX          ;;EXIT
(1)  021604  122766  000012  000002  1$:    CMPB    #12,2(SP)       ;;BRANCH IF
(1)  021612  002002                        BGE     $TYPEX          ;;<LF>
```

```
(1)  021614  105227                        INCB    (PC)+           ;;INC SPACE
(1)  021616  000000                $CHARCNT:.WORD  0               ;;COUNT
(1)  021620  000207                $TYPEX: RTS     PC
(1)                                ;;      EQUATES
(1)          000011                TAT=11
(1)          000200                TCRLF=200
(1)
4991                               ;*****************************************************************
(1)
(1)                                .SBTTL  APT COMMUNICATIONS ROUTINE
(1)  021622  112767  000001  000236 $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
(1)  021630  112767  000001  000226 $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
(1)  021636  000403                        BR      $ATYC
(1)  021640  112767  000001  000220 $ATY4:  MOVB    #1,$FFLG        ;TO ONLY REPORT FATAL ERROR
(2)  021646                        $ATYC:
(3)  021646  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  021650  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(1)  021652  105767  000206                TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
(1)  021656  001450                        BEQ     5$              ;IF NOT:  BR
(1)  021660  122767  000001  157362        CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
(1)  021666  001031                        BNE     3$              ;IF NOT:  BR
(1)  021670  132767  000100  157353        BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGES?
(1)  021676  001425                        BEQ     3$              ;IF NOT:  BR
(1)  021700  017600  000004                MOV     @4(SP),R0       ;GET MESSAGE ADDR.
(1)  021704  062766  000002  000004        ADD     #2,4(SP)                ;BUMP RETURN ADDR.
(1)  021712  005767  157312        1$:     TST     $MSGTYPE        ;SEE IF DONE W/ LAST XMISSION?
(1)  021716  001375                        BNE     1$              ;IF NOT:  WAIT
(1)  021720  010067  157320                MOV     R0,$MSGAD       ;PUT ADDR IN MAILBOX
(1)  021724  105720                2$:     TSTB    (R0)+           ;FIND END OF MESSAGE
(1)  021726  001376                        BNE     2$
(1)  021730  166700  157310                SUB     $MSGAD,R0       ;SUB START OF MESSAGE
(1)  021734  006200                        ASR     R0              ;GET MESSAGE LNGTH IN WORDS
(1)  021736  010067  157304                MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
(1)  021742  012767  000004  157260        MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MSG.
(1)  021750  000413                        BR      5$
(1)  021752  017667  000004  000016 3$:     MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
(1)  021760  062766  000002  000004        ADD     #2,4(SP)                ;BUMP RETURN ADDRESS
(3)  021766  016746  156004                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
(1)  021772  004767  177346                JSR     PC,$TYPE        ;CALL TYPE MACRO
(1)  021776  000000                4$:     .WORD   0
(1)  022000                        5$:
(1)  022000  105767  000062        10$:    TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
(1)  022004  001416                        BEQ     12$             ;IF NOT:  BR
(1)  022006  005767  157236                TST     $ENV            ;RUNNING UNDER APT?
(1)  022012  001413                        BEQ     12$             ;IF NOT:  BR
(1)  022014  005767  157210        11$:    TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
(1)  022020  001375                        BNE     11$             ;IF NOT:  WAIT
(1)  022022  017667  000004  157202        MOV     @4(SP),$FATAL   ;GET ERROR #
(1)  022030  062766  000002  000004        ADD     #2,4(SP)                ;BUMP RETURN ADDR.
(1)  022036  005267  157166                INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
(1)  022042  105067  000020        12$:    CLRB    $FFLG           ;CLEAR FATAL FLAG
(1)  022046  105067  000013                CLRB    $LFLG           ;CLEAR LOG FLAG
(1)  022052  105067  000006                CLRB    $MFLG           ;CLEAR MESSAGE FLAG
(3)  022056  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
   (3)  022060  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
   (1)  022062  000207                          RTS     PC              ;RETURN
   (1)
   (1)  022064     000          $MFLG:  .BYTE   0               ;MESSG. FLAG
   (1)  022065     000          $LFLG:  .BYTE   0               ;LOG FLAG
   (1)  022066     000          $FFLG:  .BYTE   0               ;FATAL FLAG
   (1)          022070                  .EVEN
   (1)          000200          APTSIZE=200
   (1)          000001          APTENV=001
   (1)          000100          APTSPOOL=100
   (1)          000040          APTCSUP=040
  4992                          ;*****************************************************************
   (1)
   (1)                          .SBTTL  TTY INPUT ROUTINE
   (1)
   (1)                          ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
   (1)                          ;*CALL:
   (1)                          ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
   (1)                          ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
   (1)                          ;
   (1)
   (1)  022070  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
   (1)  022072  016666  000004  000002  MOV     4(SP),2(SP)     ;;SAVE THE PS
   (1)  022100  105777  157036          1$:     TSTB    @$TKS           ;;WAIT FOR
   (1)  022104  100375                          BPL     1$              ;;A CHARACTER
   (1)  022106  117766  157032  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
   (1)  022114  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
   (1)  022122  000002                          RTI                     ;;GO BACK TO USER
   (2)                          ;*****************************************************************
   (1)                          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
   (1)                          ;*CALL:
   (1)                          ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
   (1)                          ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
   (1)                          ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
   (1)
   (1)  022124  010346          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
   (1)  022126  012703  022232          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
   (1)  022132  022703  022242          2$:     CMP     #$TTYIN+80.,R3  ;;BUFFER FULL?
   (1)  022136  101405                          BLOS    4$              ;;BR IF YES
   (1)  022140  104405                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
   (1)  022142  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
   (1)  022144  122713  000177                  CMPB    #177,(R3)       ;;IS IT A RUBOUT
   (1)  022150  001003                          BNE     3$              ;;SKIP IF NOT
   (1)  022152  104400  001224          4$:     TYPE    ,$QUES          ;;TYPE A '?'
   (1)  022156  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
   (1)  022160  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
   (1)  022164  104400  022230                  TYPE    ,9$
   (1)  022170  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
   (1)  022174  001356                          BNE     2$              ;;LOOP IF NOT RETURN
   (1)  022176  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
   (1)  022202  104400  001226                  TYPE    ,$LF            ;;TYPE A LINE FEED
   (1)  022206  012603                          MOV     (SP)+,R3        ;;RESTORE R3
   (1)  022210  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
   (1)  022212  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
```

```
(1)  022220  012766  022232  000004          MOV     #STTYIN,4(SP)
(1)  022226  000002                          RTI                     ;;RETURN
(1)  022230     000  9$:      .BYTE   0       ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  022231     000              .BYTE   0       ;;TERMINATOR
(1)  022232  000010  $TTYIN: .BLKB   8.      ;;RESERVE 8 BYTES FOR TTY INPUT
4993                 ;*****************************************************************
(1)
(1)                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
(1)
(1)                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1)                  ;*CHANGE IT TO BINARY.
(1)                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
(1)                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
(1)                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED, THE COMPLETE NUMBER MUST
(1)                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
(1)                  ;*CALL:
(1)                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
(1)                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
(1)                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
(1)
(1)  022242  011646  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
(1)  022244  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
(3)  022252  010046              MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  022254  010146              MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  022256  010246              MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(1)  022260  104406  1$:      RDLIN                   ;;READ AN ASCIZ LINE
(1)  022262  012600              MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
(1)  022264  010067  000100          MOV     R0,5$           ;;AND SAVE IT
(1)  022270  005001              CLR     R1              ;;CLEAR DATA WORD
(1)  022272  005002              CLR     R2
(1)  022274  112046  2$:      MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
(1)  022276  001420              BEQ     3$              ;;IF ZERO GET OUT
(1)  022300  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
(1)  022304  003026              BGT     4$              ;;IS AN OCTAL DIGIT
(1)  022306  122716  000067          CMPB    #'7,(SP)
(1)  022312  002423              BLT     4$
(1)  022314  006301              ASL     R1              ;;*2
(1)  022316  006102              ROL     R2
(1)  022320  006301              ASL     R1              ;;*4
(1)  022322  006102              ROL     R2
(1)  022324  006301              ASL     R1              ;;*8
(1)  022326  006102              ROL     R2
(1)  022330  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
(1)  022334  062601              ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
(1)  022336  000756              BR      2$              ;;LOOP
(1)  022340  005726  3$:      TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
(1)  022342  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
(1)  022346  010267  000026          MOV     R2,$HIOCT
(3)  022352  012602              MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  022354  012601              MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  022356  012600              MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  022360  000002              RTI                     ;;RETURN
(1)  022362  005726  4$:      TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
(1)  022364  105010              CLRB    (R0)            ;;SET A TERMINATOR
```

```
(1)  022366  104400                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
(1)  022370  000000              5$:       .WORD   0
(1)  022372  104400  001224                TYPE    .SQUES          ;;"?" "CR" & "LF"
(1)  022376  000730                        BR      1$              ;;TRY AGAIN
(1)  022400  000000              $HIOCT:   .WORD   0               ;;HIGH ORDER BITS GO HERE
4994                                      ;*****************************************************************
(1)
(1)                                       .SBTTL  TRAP DECODER
(1)
(1)                                       ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)                                       ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)                                       ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1)                                       ;*GO TO THAT ROUTINE.
(1)
(1)  022402  010046              $TRAP:    MOV     R0,-(SP)        ;;SAVE R0
(1)  022404  016600  000002                MOV     2(SP),R0        ;;GET TRAP ADDRESS
(1)  022410  005740                        TST     -(R0)           ;;BACKUP BY 2
(1)  022412  111000                        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
(1)  022414  006300                        ASL     R0              ;;POSITION FOR INDEXING
(1)  022416  016000  022424                MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
(1)  022422  000200                        RTS     R0              ;;GO TO ROUTINE
(1)
(3)
(3)                                       .SBTTL  TRAP TABLE
(3)
(3)                                       ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3)                                       ;*BY THE "TRAP" INSTRUCTION.
(3)
(3)                                       ;       ROUTINE
(3)                                       ;       -------
(3)  022424                               $TRPAD:
(3)  022424  021344                                $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
(3)  022426  020716                                $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(3)  022430  020672                                $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
(3)  022432  020732                                $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
(3)  022434  021120                                $TYPDS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
(3)  022436  022070                                $RDCHR  ;;CALL=RDCHR    TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
(3)  022440  022124                                $RDLIN  ;;CALL=RDLIN    TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
(3)  022442  022242                                $RDOCT  ;;CALL=RDOCT    TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
4995                                      ;*****************************************************************
(1)
(1)                                       .SBTTL  POWER DOWN AND UP ROUTINES
(1)
(1)                                       ;POWER DOWN ROUTINE
(1)  022444  012737  022572  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
(1)  022452  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
(3)  022460  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  022462  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  022464  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  022466  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  022470  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
(3)  022472  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  022474  010667  000076                MOV     SP,$SAVR6       ;;SAVE SP
(1)  022500  012737  022512  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
```

```
(1)  022506  000000                          HALT
(1)  022510  000776                          BR      .-2             ;;HANG UP
(1)
(1)                                  ;POWER UP ROUTINE
(1)  022512  016706  000060          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
(1)  022516  005067  000054                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
(1)  022522  005267  000050          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
(1)  022526  001375                          BNE     1$              ;;OF  WORD
(3)  022530  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  022532  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
(3)  022534  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  022536  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  022540  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  022542  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  022544  012737  022444  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1)  022552  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
(1)  022560  104400                          TYPE                    ;REPORT THE POWER FAILURE
(1)  022562  022600                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
(1)  022564  012716                          MOV     (PC)+,(SP)      ;;RESTART AT RSTRTA
(1)  022566  002612                  $PWRAD: .WORD   RSTRTA          ;;RESTART ADDRESS
(1)  022570  000002                          RTI
(1)  022572  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
(1)  022574  000776                          BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
(1)  022576  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
(1)  022600  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
(1)  022606  000122
(1)                                          .EVEN
```

```
                                        ;*****************************************************************
                                        ;COMMON DH11 SERVICE ROUTINES
                                        ;*****************************************************************

                                        ;THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
                                        ;OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN

022610  012701  033436          LDTBF1: MOV     #TBUF,R1        ;POINT TO START OF BUFFER
022614  005002                  CLR     R2              ;INIT DATA BYTE GENERATOR
022616  112221                  1$:     MOVB    R2,(R1)+        ;LOAD ONE CHAR
022620  005202                  INC     R2              ;GENERATE NEXT CHAR
022622  022702  000400          CMP     #400,R2         ;LOADED 256(10) BYTES
022626  001373                  BNE     1$              ;BR IF NOT
022630  000207                  RTS     PC              ;RETURN TO START TESTING

                                        ;THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
                                        ;USING A "DH1" HEADER

022632  004767  001476          SUER1:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
022636  116700  156240                  MOVB    $TSTNM,R0       ;SAVE THE TEST NO.
022642  010067  156312                  MOV     R0,$REG0        ;SAVE THE TEST NO. FOR ERROR PRINT
022646  010167  156310                  MOV     R1,$REG1        ;SAVE THE DH11 ADDR
022652  010267  156306                  MOV     R2,$REG2        ;SAVE THE REG ADDRESS
022656  010667  156312                  MOV     R6,$REG6        ;SAVE THE SP
022662  062767  000002  156304          ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
022670  000207                  RTS     PC              ;RETURN TO CALLING ROUTINE

                                        ;THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
                                        ;SAVE THE ERROR INFORMATION IN "DT2"

022672  004767  001436          SUER2:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
022676  116700  156200          SUER2A: MOVB    $TSTNM,R0       ;GET THE TEST NO.
022702  010067  156252                  MOV     R0,$REG0        ;SAVE THE REGISTERS-TEST#
022706  010167  156250                  MOV     R1,$REG1        ;SAVE THE DH ADDRESS
022712  010267  156246                  MOV     R2,$REG2        ;SAVE THE REGISTER ADDRESS
022716  010367  156244                  MOV     R3,$REG3        ;SAVE THE WAS DATA
022722  010467  156242                  MOV     R4,$REG4        ;SAVE THE S/B DATA
022726  010667  156240                  MOV     R6,$REG6        ;SAVE THE STACK POINTER
022732  062767  000002  156234          ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
022740  000207                  RTS     PC              ;RETURN TO REPORT ERROR

                                        ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                        ;BUS ERROR AND RSVD INSTR ERROR ROUTINES

022742  010067  156212          SUER3:  MOV     R0,$REG0        ;SAVE THE REGS
022746  010167  156210                  MOV     R1,$REG1
022752  010267  156206                  MOV     R2,$REG2
022756  000207                  RTS     PC              ;RETURN TO REPORT ERROR

                                        ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                        ;CAR/BCR MEMORY PATTERNS TESTS

022760  005067  156214          SUER4:  CLR     $TMP0           ;SAVE THE LINE NO. WRITTEN
022764  116767  002156  156206          MOVB    LINEA,$TMP0
```

```
                                                MOVB    $TSTNM,R0       ;SAVE THE TEST NUMBER
                                                MOV     R0,$REG0        ;SAVE THE REGISTER INFORMATION
                                                MOV     R1,$REG1
                                                MOV     R2,$REG2
                                                MOV     R3,$REG3
                                                MOV     R4,$REG4
                                                RTS     PC              ;RETURN TO PATTERNS TEST

                                ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
                                ;VALUE OF THE LINE SELECTION PARAMETER

                                        ;CALLING SEQUENCE:

                                        ;JSR     PC,SELINE      ;CALL THE ROUTINE
                                        ;BR      1$             ;EXIT BRANCH-ROUTINE MOVES THE RETURN
                                                                ;PC AROUND THIS BR IF MORE LINES ARE
                                                                ;YET TO BE TESTED

023024  105767  002115          SELINE: TSTB    LINE+1          ;FIRST TIME THROUGH FOR ANY TEST ?
023030  001010                          BNE     1$              ;BR IF NOT
023032  105167  002107                  COMB    LINE+1          ;SET ENTRY FLAG
023036  012767  000001  001346          MOV     #1,LINMSK       ;INIT SELECT TEST MASK TO TEST LINE 00
023044  105067  002074                  CLRB    LINE            ;START WITH LINE #00
023050  000405                          BR      2$              ;GO TEST FOR LINE #00
023052  105267  002066          1$:     INCB    LINE            ;GENERATE NEW LINE NO.
023056  006367  001330                  ASL     LINMSK          ;SHIFT SELECT MASK TO TEST NXT LINE
023062  001407                          BEQ     3$              ;RETURN TO EXIT BRANCH - ALL LINES DONE
023064  036767  001322  001316  2$:     BIT     LINMSK,LINSEL   ;IS THE LINE SELECTED FOR TEST ??
023072  001767                          BEQ     1$              ;BR IF NOT
023074  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND EXIT BRANCH
023100  000402                          BR      4$              ;RETURN TO TEST SELECTED LINE
023102  005067  002036          3$:     CLR     LINE            ;INIT ENTRY FLAG AND LINE NO. TO 000
023106  142777  000017  001264  4$:     BICB    #17,@DHADR      ;INIT LINE SELECT BITS IN "SCR"
023114  000207                          RTS     PC              ;RETURN TO CALLING TEST

                                ;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
                                ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
                                ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING

                                        ;CALLING SEQUENCE

                                        ;JSR     R5,SUNUM       ;CALL TO THIS ROUTINE
                                        ;ADDR1                  ;ADDRESS OF THE NUMBER TO BE CONVERTED
                                        ;ADDR2                  ;ADDRESS OF THE MSG BUFFER SLOT

023116                          SUNUM:
(2) 023116  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(2) 023120  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(2) 023122  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023124  012500                          MOV     (R5)+,R0        ;GET ADDRESS OF NUMBER
023126  012501                          MOV     (R5)+,R1        ;GET MSG BUFFER ADDR
023130  111000                          MOVB    (R0),R0         ;GET NO. TO BE CONVERTED
023132  010002                          MOV     R0,R2           ;SAVE IT IN R2
023134  006202                          ASR     R2              ;SHIFT MSD TO LSD POSITION
```

```
                                        ASR     R2
                                        ASR     R2
023136  006202                          ASR     R2
023140  006202                          ASR     R2
023142  042702  177770                  BIC     #177770,R2      ;CLR JUNK BITS
023146  062702  000060                  ADD     #60,R2          ;MAKE IT ASCII
023152  110221                          MOVB    R2,(R1)+        ;PUT IT IN MSG BUFFER
023154  042700  177770                  BIC     #177770,R0      ;CLR JUNK FROM LSD
023160  062700  000060                  ADD     #60,R0          ;MAKE IT ASCII
023164  110011                          MOVB    R0,(R1)         ;PUT LSD IN THE BUFFER
023166  012602                          MOV     (SP)+,R2        ;POP STACK INTO R2
023170  012601                          MOV     (SP)+,R1        ;POP STACK INTO R1
023172  012600                          MOV     (SP)+,R0        ;POP STACK INTO R0
023174  000205                          RTS     R5              ;RETURN TO CALLER

                                ;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
                                ;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1

023176  005067  156014          CLCABC: CLR     $TMP7           ;INIT A COUNTER
023202  116711  156010          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
023206  005061  000006                  CLR     CAR(R1)         ;CLEAR A CAR LOCATION
023212  005061  000010                  CLR     BCR(R1)         ;CLEAR A BCR LOCATION
023216  005267  155774                  INC     $TMP7           ;GENERATE NEW LINE NO.
023222  022767  000020  155766          CMP     #20,$TMP7       ;DONE ALL LINES ?
023230  001364                          BNE     1$              ;BR IF NOT
023232  142711  000017                  BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
023236  000207                          RTS     PC              ;RETURN TO CALLER

                                ;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
                                ;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1

023240  005067  155752          LDBCR:  CLR     $TMP7           ;INIT A COUNTER
023244  116711  155746          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
023250  012761  177777  000010          MOV     #-1,BCR(R1)     ;LOAD BCR LOC. WITH 177777
023256  005267  155734                  INC     $TMP7           ;GENERATE NEXT LINE NO.
023262  022767  000020  155726          CMP     #20,$TMP7       ;DONE ALL LINES ?
023270  001365                          BNE     1$              ;BR IF NOT
023272  142711  000017                  BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
023276  000207                          RTS     PC              ;RETURN TO CALLER

                                ;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS

023300  012767  000020  155702  SUPPAR: MOV     #20,$TMP4       ;SET UP FOR 16. LINES
023306  105011                          CLRB    (R1)            ;INIT SCR TO START AT LINE 00
023310  005002                          CLR     R2              ;INIT INDEX REGISTER FOR RBUF (EVEN)
023312  012703  000200                  MOV     #200,R3         ;SET UP CONSTANT
023316  012704  000001                  MOV     #1,R4           ;INIT INDEX REG FOR RBUF (ODD)
023322  012761  033436  000006  1$:     MOV     #TBUF,CAR(R1)   ;LOAD BUS ADDRESS REWG
023330  016761  155660  000010          MOV     $TMP6,BCR(R1)   ;LOAD BYTE COUNT REG
023336  016761  155650  000004          MOV     $TMP5,LPR(R1)   ;LOAD LINE PARAMETERS
023344  105062  032436                  CLRB    RBUF(R2)        ;INIT DATA BYTE IN RBUF TO START AT 000
023350  110364  032436                  MOVB    R3,RBUF(R4)     ;SET CONSTANT IN HIGH BYTE
023354  005211                          INC     (R1)            ;SELECT NEXT LINE
023356  005203                          INC     R3              ;GENERATE NEW CONSTANT
023360  062702  000002                  ADD     #2,R2           ;UPDATE POINTERS TO RBUF (EVEN/ODD)
023364  062704  000002                  ADD     #2,R4
```

```
                                        DEC     STMP4           ;COUNT ONE LINE SETUP
                                        BNE     1$              ;BR TILL ALL 16. SET UP
                                        MOV     #MULPTB,R4      ;SET UP TABLE POINTER
                                2$:     MOV     STMP6,(R4)+     ;SET UP BYTE COUNT ENTRY
                                        CMP     #MULPTB+40,R4   ;SET UP ALL COUNTS ?
                                        BNE     2$              ;BR IF NOT
                                        CLRB    (R1)            ;INIT SCR TO SELECT LINE 00
                                        RTS     PC              ;RETURN TO PARITY TEST

                                ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
                                ;TELETYPE

                                INPARA: TYPE
                                        VCWC                    ;"ASK FOR NO. WORDS BETWEEN VECTORS"
                                        RDOCT                   ;READ OCTAL NO. FM TTY
                                        MOV     (SP)+,R0        ;GET THE NO. HE TYPED
                                        BEQ     3$              ;BR IF HE TYPED <CR>
                                        CMP     #4,R0           ;FOUR WORDS BETWEEN VECTORS ?
                                        BEQ     2$              ;BR IF YES
                                        CMP     #10,R0          ;8. WORDS BETWEEN VECTORS ??
                                        BEQ     3$              ;BR IF YES
                                        BR      INPARA          ;ASK ALL OVER AGAIN
                                2$:     MOV     #10,R0          ;SET UP CONSTANT IN R0 FOUR 4 WORDS
                                        BR      4$              ;CONTINUE
                                3$:     MOV     #20,R0          ;SET UP CONSTANT FOR 8. WORDS
                                4$:     RTS     PC              ;RETURN TO CALLER


                                INPARC: MOV     #-1,R0          ;SET FLAG IN R0
                                        JMP     BEGINA          ;GO ASK FOR SELECT PARAMETER

                                INPARX: MOV     #-1,VCFLG       ;SET SETUP FLAG
                                        JMP     BEGINA          ;GO START UP

                                INPAR:  TYPE                    ;ASK FOR DEVICE ADDRESS
                                        INMSG1
                                        RDOCT                   ;READ IN WHAT IS TYPED
                                        MOV     (SP)+,R1        ;GET THE NO. HE TYPED
                                        BEQ     INPAR1          ;BR IF DEFAULT
                                        JSR     PC,CHKADR       ;GO CHECK VALIDITY OF THE ADDR
                                        BR      INPAR           ;ERROR BRANCH

                                INPAR1: TYPE                    ;ASK FOR VECTOR ADDRESS
                                        INMSG2
                                        RDOCT                   ;READ IN WHAT HE TYPES
                                        MOV     (SP)+,R1        ;GET THE ADDRESS
                                        BEQ     INPAR3          ;BR IF DEFAULT
                                        JSR     PC,CHKVCT       ;GO CHECK VALIDITY OF VECTOR
                                        BR      INPAR1          ;ERROR BRANCH

                                INPAR3: TYPE                    ;ASK FOR DEVICE SELECTION PARAMETER
                                        INMSG3
                                        RDOCT                   ;READ IN WHAT HE TYPES
```

```
                                        MOV     (SP)+,R1        ;GET THE SELECT PARAMETER
                                        BEQ     INPAR4          ;BR IF  DEFAULT
                                        MOV     R1,DHSEL        ;SET UP DH11 SELECTION PARAMETER

                        INPAR4: TYPE                            ;ASK FOR LINE SELECT PARAMETER
                                INMSG6
                                RDOCT                           ;GET WHAT HE TYPES
                                MOV     (SP)+,R1                ;GET PARAMETER
                                BEQ     1$                      ;BR IF DEFAULT
                                MOV     R1,LINSEL               ;SET UP LINE SELECT PARAMETER
                                BR      2$                      ;CONTINUE
                        1$:     MOV     #-1,LINSEL              ;SET UP DEFAULT (ALL LINES)
                        2$:     TST     @SWR                    ;HALT AFTER SET UP ??
                                BPL     EXPAR                   ;BR IF  NOT
                                TYPE                            ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
                                INMSG7
                                HALT                            ;DEPRESS CONTINUE TO RESUME TESTING
                        EXPAR:  JMP     START2                  ;GO START UP THE PROGRAM


                        CHKADR: CMP     R1,#160020              ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
                                BGE     1$                      ;BR IF YES
                                BR      4$                      ;BR IF NOT
                        1$:     CMP     R1,#160420              ;IS IT BELOW THE HIGH LIMIT?
                                BLT     2$                      ;BR IF YES
                                BR      4$                      ;BR IF NOT
                        2$:     BIT     #17,R1                  ;CORRECT BOUNDARY ?
                                BNE     4$                      ;BR IF NOT
                                ADD     #2,(SP)                 ;MOVE RETURN PC AROUND ERROR BRANCH
                                MOV     #DHADTB,R2              ;POINT TO BEGIN OF ADDR TABLE
                        3$:     MOV     R1,(R2)+                ;SETR UP A TABLE ENTRY
                                ADD     #20,R1                  ;GENERATE NEXT DH11 ADDR
                                CMP     #DHADTB+40,R2           ;END OF TABLE ?
                                BNE     3$                      ;BR IF NOT
                                BR      5$                      ;RETURN TO INPUT ROUTINES
                        4$:     TYPE                            ;TELL HIM HE GOOFED
                                INMSG4
                        5$:     RTS     PC                      ;RETURN TO INPUT ROUTINES


                        CHKVCT: CMP     R1,#300                 ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
                                BGE     1$                      ;BR IF YES
                                BR      4$                      ;BR IF NOT
                        1$:     CMP     R1,#1000                ;IS IT BELOW THE HIGH LIMIT?
                                BLT     2$                      ;BR IF YES
                                BR      4$                      ;BR IF NOT
                        2$:     BIT     #7,R1                   ;CORRECT BOUNDARY ?
                                BNE     4$                      ;BR IF NOT
                                ADD     #2,(SP)                 ;MOVE RETURN PC AROUND ERROR BRANCH
                                MOV     #DHVCTB,R2              ;POINT TO BEGIN OF VECTOR TABLE
                        3$:     MOV     R1,(R2)+                ;SETR UP A TABLE ENTRY
                                ADD     R0,R1                   ;GENERATE NEXT DH11 ADDR
                                CMP     #DHVCTB+40,R2           ;END OF TABLE ?
                                BNE     3$                      ;BR IF NOT
                                BR      5$                      ;RETURN TO INPUT ROUTINES
```

```
 5263  023766  104400                    4$:     TYPE                    ;TELL HIM HE GOOFED
 5264  023770  032120                            INMSG5
 5265  023772  000207                    5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
 5266
 5267                                    ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
 5268
 5269  023774  012767  000340  155176    BUSER:  MOV     #340,$TMP0      ;SAVE THE PSW
 5270  024002  010667  155166            MOV     SP,$REG6        ;SAVE THE SP
 5271  024006  012601                    MOV     (SP)+,R1        ;GET THE TRAP PC
 5272  024010  012602                    MOV     (SP)+,R2        ;GET THE TRAP PSW
 5273  024012  116700  155064            MOVB    $TSTNM,R0       ;GET TEST NO.
 5274  024016  012706  001100            MOV     #STACK,SP       ;RESET THE STACK POINTER
 5275  024022  004767  176714            JSR     PC,SUER3        ;GO SET UP ERROR INFO
 5276  024026  012767  024036  155054    MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
 5277  024034  104027                    ERROR   27              ;UNEXPECTED BUS ERROR TRAP
 5278  024036  000005            1$:     RESET                   ;PREPARE TO RESTART
 5279  024040  004767  000240            JSR     PC,CHPS1        ;GO CLEAR PSW
 5280  024044  000167  156532            JMP     REST1           ;GO RESTART THE PROGRAM
 5281
 5282  024050  012767  000340  155122    RESERR: MOV     #340,$TMP0      ;SAVE THE PSW
 5283  024056  010667  155112            MOV     SP,$REG6        ;SAVE THE SP
 5284  024062  012601                    MOV     (SP)+,R1        ;GET THE TRAP PC
 5285  024064  012602                    MOV     (SP)+,R2        ;GET THE TRAP PSW
 5286  024066  116700  155010            MOVB    $TSTNM,R0       ;GET TEST NO.
 5287  024072  012706  001100            MOV     #STACK,SP       ;RESET THE STACK POINTER
 5288  024076  004767  176640            JSR     PC,SUER3        ;GO SET UP ERROR INFO
 5289  024102  012767  024112  155000    MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
 5290  024110  104030                    ERROR   30              ;UNEXPECTED RSVD INSTR ERROR TRAP
 5291  024112  000005            1$:     RESET                   ;PREPARE TO RESTART
 5292  024114  004767  000164            JSR     PC,CHPS1        ;GO CLEAR PSW
 5293  024120  000167  156456            JMP     REST1           ;GO RESTART THE PROGRAM
 5294
 5295                                    ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
 5296                                    ;CATCHER IN THE DH11 VECTOR
 5297
 5298  024124  016703  000252            RESTRP: MOV     DHVCT,R3        ;GET VECTOR ADDRESS
 5299  024130  010313                    MOV     R3,(R3)         ;RESTORE THE TRAP CATCHER
 5300  024132  062723  000002            ADD     #2,(R3)+
 5301  024136  005023                    CLR     (R3)+
 5302  024140  010313                    MOV     R3,(R3)
 5303  024142  062723  000002            ADD     #2,(R3)+
 5304  024146  005023                    CLR     (R3)+
 5305  024150  000207                    RTS     PC              ;RETURN TO CALLING TEST
 5306
 5307                                    ;THIS ROUTINE CALLED BY ANY TEST THAT NEEDS A TIMING WAIT LOOP
 5308                                    ;"TIMEA" IS INITIALIZED BY THE CALLING ROUTINE TO THE MINIMUM REQUIRED
 5309                                    ;VALUE AND "TIMEB" IS CLEARED TO 000000. IF A TIME OUT OCCURS THIS
 5310                                    ;ROUTINE WILL MOVE THE RETURN PC AROUND THE "LOOP" BRANCH BACK IN
 5311                                    ;THE ROUTINE THAT CALLED IT TO ALLOW REPORTING AN ERROR MESSAGE
 5312
 5313  024152  005267  001016            TIMEIT: INC     TIMEB           ;COUNT B
 5314  024156  001005                    BNE     1$              ;BR IF NOT ZERO
 5315  024160  005367  001006            DEC     TIMEA           ;COUNT TIME A
 5316  024164  001002                    BNE     1$              ;BR IF NO TIMEOUT
```

```
5317  024166  062716  000002                    ADD     #2,(SP)         ;MOVE RETURN PC TO ALLOW ERROR REPORT
5318  024172  000207                    1$:     RTS     PC              ;RETURN TO THE CALLING TEST
5319
5320                                             ;THIS ROUTINE CALLED BY THE AUTO ECHO TEST TO SET UP FOR TRANSFERRING
5321                                             ;A BINARY COUNT TEST PATTERN ON ALL LINES
5322
5323  024174  012767  000020  155012  SETALL: MOV     #20,$TMP6       ;SET UP SIXTEEN LINES
5324  024202  005002                    CLR     R2              ;INIT A TABLE INDEX REG
5325  024204  012703  000200                    MOV     #200,R3         ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
5326  024210  012704  000001                    MOV     #1,R4           ;SET UP INDEX REG TO 000 BYTES
5327  024214  005011                    CLR     (R1)            ;START WITH LINE 00
5328  024216  012761  033436  000004  1$:      MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDR REG
5329  024224  012761  177400  000010          MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT REG
5330  024232  012761  031403  000004          MOV     #31403,LPR(R1)  ;SET UP FOR 4800 BAUD/8 BIT CHARS
5331  024240  105062  032436                  CLRB    RBUF(R2)        ;START WITH DATA CHAR OF 000
5332  024244  110364  032436                  MOVB    R3,RBUF(R4)     ;SET UP HIGH BYTE OF EXPECTED DATA
5333  024250  005211                    INC     (R1)            ;GEN NEW LINE NO. IN SCR
5334  024252  005203                    INC     R3              ;UPDATE THE POINTERS AND DATA
5335  024254  062702  000002                    ADD     #2,R2
5336  024260  062704  000002                    ADD     #2,R4
5337  024264  005367  154724                    DEC     $TMP6           ;COUNT ONE LINE DONE
5338  024270  001352                    BNE     1$              ;BR TIL ALL 16 SET UP
5339  024272  016767  000112  000356          MOV     LINSEL,LINACT   ;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
5340  024300  005011                    CLR     (R1)            ;PUT SCR REG BACK TO LINE 00
5341  024302  000207                    RTS     PC              ;RETURN TO AUTO ECHO TEST
5342
5343
5344                                             ;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
5345                                             ;TO BE LSI11 COMPATIBLE
5346
5347  024304  012746  000000          CHPS1:  MOV     #0,-(SP)        ;NEW PSW
5348  024310  012746  024316                  MOV     #1$,-(SP)       ;NEW PC
5349  024314  000002                    RTI                     ;CHANGE PSW
5350  024316  000207                    1$:     RTS     PC              ;RETURN TO CALLING TEST
5351
5352                                             ;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSW
5353                                             ;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTRS
5354
5355  024320  012746  000340          CHPS2:  MOV     #340,-(SP)      ;NEW PSW
5356  024324  012746  024332                  MOV     #1$,-(SP)       ;NEW PC
5357  024330  000002                    RTI                     ;CHANGE THE PSW
5358  024332  000207                    1$:     RTS     PC              ;RETURN TO CALLING TEST
5359
5360                                             ;THIS ROUTINE IS ALSO FOR LSI11 COMPATIBILITY AND IT IS CALLED
5361                                             ;TO SAVE THE PSW IN "$TMP0"
5362
5363  024334  005046                    SAPS:   CLR     -(SP)           ;TEMP STORAGE TO SAVE PSW
5364  024336  016746  153472                    MOV     34,-(SP)        ;SAVE TRAP VECTOR POINTER
5365  024342  012767  024352  153464          MOV     #1$,34          ;GO TO 1$ ON TRAP
5366  024350  104400                    TRAP                    ;GO TO IT
5367  024352  016666  000002  000006  1$:      MOV     2(SP),6(SP)     ;GET PSW SAVED
5368  024360  012716  024366                  MOV     #2$,(SP)        ;GO TO 2$ ON RTI
5369  024364  000002                    RTI
5370  024366  012667  153442          2$:     MOV     (SP)+,34        ;RESTORE VECTOR
```

```
 5371  024372  012667  154602                  MOV     (SP)+,STMPO     ;FINALLY SAVE PSW IN STMPO
 5372  024376  000207                          RTS     PC
 5373
```

```
                                ;*****************************************************************
                                ;ADDITIONAL PROGRAM CONSTANTS AND VARIABLES
                                ;*****************************************************************

            000002                      NRC=2           ;INDEX CONST. TO ACCESS NEXT RCVD CHAR REG
            000004                      LPR=4           ;INDEX CONST. TO ACCESS LINE PARAMETER REG.
            000006                      CAR=6           ;INDEX CONST. TO ACCESS CURRENT ADDRESS REG.
            000010                      BCR=10          ;INDEX CONST. TO ACCESS BYTE COUNT REG.
            000012                      BAR=12          ;INDEX CONST. TO ACCESS BUFFER ACTIVE REG.
            000014                      BKR=14          ;INDEX CONST. TO ACCESS BREAK CONTROL REG.
            000016                      SSR=16          ;INDEX CONST. TO ACCESS SILO STATUS REG.

024400  000000  DHADR:  0               ;HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
024402  000000  DHVCT:  0               ;HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
024404  000000  SELMSK: 0               ;BIT TST MARKER FOR SELECTING DH11'S
024406  000003  DHSEL:  3               ;SPECIFIES DH11'S SELECTED FOR TEST
024410  177777  LINSEL: 177777          ;SPECIFIES LINES TO TEST
024412  000000  LINMSK: 0               ;MARKER USED TO TEST FOR LINES TO TEST
024414  000000  LMSK1:  0               ;ALTERNATE MARKER TO SUPPORT THE
                                        ;SELECT LINES FEATURE
024416  000004  MSTCLR: .BLKW   4       ;FOUR WORD ADDRESS TABLE USED BY THE TEST THAT
                                        ;CHECKS OPERATION OF "MASTER CLR"

024426  177777  PATRNA: 177777          ;BIT PATTERNS USED WITH "CAR" AND "BCR" TESTS
024430  125252          125252
024432  052525          052525
024434  000000          000000          ;TABLE TERMINATOR

024436  000060  PATRNB: 60              ;BIT PATTERNS USED IN "CAR" MEM EXT BIT TEST
024440  000300          300
024442  000020          20
024444  000100          100
024446  000040          40
024450  000200          200
024452  000000          0               ;TABLE TERMINATOR
024454  000000          0               ;TABLE TERMINATOR

                                ;THIS TABLE STORES THE BYTE COUNT AND LINE PARAMETERS FOR THE
                                ;8 SUBTESTS IN THE MULTILINE PARITY/DATA TEST

024456  177400  PRTYTB: -400                    ;256 CHARS
024460  027363          27363                   ;2400 BAUD - ODD PARITY - 8 BITS
024462  177400          -400                    ;256 CHARS
024464  027323          27323                   ;2400 BAUD - EVEN PARITY - 8 BITS
024466  177600          -200                    ;128 CHARS
024470  027362          27362                   ;2400 BAUD - ODD PARITY - 7 BITS
024472  177600          -200                    ;128 CHARS
024474  027322          27322                   ;2400 BAUD - EVEN PARITY - 7 BITS
024476  177700          -100                    ;64 CHARS
024500  027361          27361                   ;2400 BAUD - ODD PARITY - 6 BITS
024502  177700          -100                    ;64 CHARS
024504  027321          27321                   ;2400 BAUD - EVEN PARITY - 6 BITS
024506  177740          -40                     ;32 CHARS
024510  027360          27360                   ;2400 BAUD - ODD PARITY - 5 BITS
```

```
                024512  177740                  -40                     ;32 CHARS
                024514  027320                  27320                   ;2400 BAUD - EVEN PARITY - 5 BITS
                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                        ;TEST (ALL 1'S DATA TABLE)
                024516  100377          RETAB:  100377          ;TEST DATA FOR LINE 00
                024520  100777                  100777          ;TEST DATA FOR LINE 01
                024522  101377                  101377
                024524  101777                  101777
                024526  102377                  102377
                024530  102777                  102777
                024532  103377                  103377
                024534  103777                  103777
                024536  104377                  104377
                024540  104777                  104777
                024542  105377                  105377
                024544  105777                  105777
                024546  106377                  106377
                024550  106777                  106777
                024552  107377                  107377
                024554  107777                  107777          ;TEST DATA FOR LINE 17

                                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                        ;TEST (ALL 0'S DATA TABLE)
                024556  100000          RETAB0: 100000          ;TEST DATA FOR LINE 00
                024560  100400                  100400          ;TEST DATA FOR LINE 01
                024562  101000                  101000
                024564  101400                  101400
                024566  102000                  102000
                024570  102400                  102400
                024572  103000                  103000
                024574  103400                  103400
                024576  104000                  104000
                024600  104400                  104400
                024602  105000                  105000
                024604  105400                  105400
                024606  106000                  106000
                024610  106400                  106400
                024612  107000                  107000
                024614  107400                  107400          ;TEST DATA FOR LINE 17

                                        ;THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
                                        ;LINE IS DONE

                024616  000001          LINBIT: BIT00                   ;DEACTIVATE LINE 00
                024620  000002                  BIT01                   ;DEACTIVATE LINE 01
                024622  000004                  BIT02
                024624  000010                  BIT03
                024626  000020                  BIT04
                024630  000040                  BIT05
                024632  000100                  BIT06
                024634  000200                  BIT07
```

```
5483  024636  000400                        BIT08
5484  024640  001000                        BIT09
5485  024642  002000                        BIT10
5486  024644  004000                        BIT11
5487  024646  010000                        BIT12
5488  024650  020000                        BIT13
5489  024652  040000                        BIT14
5490  024654  100000                        BIT15                   ;DEACTIVATE LINE 17
5491
5492  024656  000000              LINACT: 0                         ;MAINTAINS STATUS OF ACTIVE LINES
5493                                                                ;DURING AUTO ECHO TEST 2
5494
5495
5496                              ;THIS TABLE CONTAINS 16. COUNTERS USED BYN THE MULTI-LINE
5497                              ;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED
5498
5499  024660  000020              MULPTB: .BLKW   16.               ;SIXTEEN WORD COUNTERS TABLE
5500
5501                              ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
5502                              ;TEST
5503
5504  024720  120000              BRKTAB: 120000          ;TEST DATA FOR LINE 00
5505  024722  120400                      120400          ;TEST DATA FOR LINE 01
5506  024724  121000                      121000
5507  024726  121400                      121400
5508  024730  122000                      122000
5509  024732  122400                      122400
5510  024734  123000                      123000
5511  024736  123400                      123400
5512  024740  124000                      124000
5513  024742  124400                      124400
5514  024744  125000                      125000
5515  024746  125400                      125400
5516  024750  126000                      126000
5517  024752  126400                      126400
5518  024754  127000                      127000
5519  024756  127400                      127400          ;TEST DATA FOR LINE 17
5520
5521  024760  131177              RGMSK1: 131177          ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
5522  024762  046600              RGMSK2: 46600           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
5523  024764  177767              RGMSK3: 177767          ;MASK TO SPECIFY R/W BITS IN "LPR"
5524  024766  177777              RGMSK4: 177777          ;MASK TO SPECIFY R/W BITS IN "BKR"
5525  024770  100077              RGMSK5: 100077          ;MASK TO SPECIFY R/W BITS IN "SSR"
5526  024772  042200              RGMSK6: 42200           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
5527  024774  030100              INTMSK: 30100           ;MASK USED TO SELECT INTR BITS TO TEST
5528
5529                              ;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
5530                              ;SIXTEEN DH11'S
5531
5532  024776  160020              DHADTB: 160020          ;ADDRESS OF FIRST DH11
5533  025000  160040                      160040          ;ADDRESS OF SECOND DH11
5534  025002  160060                      160060
5535  025004  160100                      160100
5536  025006  160120                      160120
```

```
5537  025010  160140                          160140
5538  025012  160160                          160160
5539  025014  160200                          160200
5540  025016  160220                          160220
5541  025020  160240                          160240
5542  025022  160260                          160260
5543  025024  160300                          160300
5544  025026  160320                          160320
5545  025030  160340                          160340
5546  025032  160360                          160360
5547  025034  160400                          160400          ;ADDRESS OF THE LAST DH11
5548
5549                                  ;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
5550                                  ;TO SIXTEEN DH11'S
5551
5552  025036  000330                  DHVCTB: 330             ;ADDRESS OF VECTOR FOR FIRST DH11
5553  025040  000350                          350             ;ADDRESS OF VECTOR FOR SECOND DH11
5554  025042  000370                          370
5555  025044  000410                          410
5556  025046  000430                          430
5557  025050  000450                          450
5558  025052  000470                          470
5559  025054  000510                          510
5560  025056  000530                          530
5561  025060  000550                          550
5562  025062  000570                          570
5563  025064  000610                          610
5564  025066  000630                          630
5565  025070  000650                          650
5566  025072  000670                          670
5567  025074  000710                          710             ;ADDRESS OF VECTOR FOR LAST DH11
5568
5569  025076  000000                  VCFLG:  0               ;VECTOR SET UP FLAGG
5570
5571
5572                                  ;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
5573                                  ;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
5574                                  ;AND THE XMTTR LEVEL IN THE HIGH BYTE
5575
5576  025100  120240                  BRLVL:  120240          ;BRLEVELS FOR FIRST DH11
5577  025102  120240                          120240          ;BR LEVELS FOR SECOND DH11
5578  025104  120240                          120240
5579  025106  120240                          120240
5580  025110  120240                          120240
5581  025112  120240                          120240
5582  025114  120240                          120240
5583  025116  120240                          120240
5584  025120  120240                          120240
5585  025122  120240                          120240
5586  025124  120240                          120240
5587  025126  120240                          120240
5588  025130  120240                          120240
5589  025132  120240                          120240
5590  025134  120240                          120240
```

```
                                    120240          ;BR LEVELS FOR LAST DH11
025140  000          DHRLVL: .BYTE   0               ;BR LEVEL FOR RCVR
025141  000          DHTLVL: .BYTE   0               ;BR LEVEL FOR XMITTER
025142  000000       DHNUM:  0                       ;CONTAINS NUMBER OF THE DH11 UNDER TEST
025144  000000       LINE:   0                       ;CONTINES NUMBER OF THE LINE UNDER TEST
025146  000000       LINEA:  0                       ;LOCATION TO SAVE LINE NUMBER
                     ;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS

025150  000000       ADPTR:  0                       ;POINTS TO ADDRESS TABLE
025152  000000       VCPTR:  0                       ;POINTS TO VECTOR TABLE
025154  000000       BRPTR:  0                       ;POINTS TO BR LEVEL TABLE

025156  000000       TDATA1: 0                       ;DATA BUFFER FOR BASIC DATA TEST
025160  000037       TDATA2: 37                      ;TEST DATA FOR FIVE BIT CHAR
025162  000077               77                      ;TEST DATA FOR SIX BIT CHAR
025164  000177               177                     ;TEST DATA FOR SEVEN BIT CHAR
025166  000377               377                     ;TEST DATA FOR EIGHT BIT CHAR
025170  000000       TITFLG: 0                       ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
025172  000000       TIMEA:  0                       ;GENERAL PURPOSE TIMERS
025174  000000       TIMEB:  0
025176  000000       TIMEC:  0                       ;TIMER FOR TIMING TESTS
025200  000000       TNULL:  0                       ;CONTAINS TWO NULL CHARS USED BY BREAK TEST
```

```
                                        ;*****************************************************************
                                        ;ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS
                                        ;*****************************************************************

                                        ;INFORMATION FOR MESSAGE 1

025202  044104  030461  051040  EM1:    .ASCIZ  'DH11 REGISTER REFERENCE CAUSED TIMEOUT'
025210  043505  051511  042524
025216  020122  042522  042506
025224  042522  041516  020105
025232  040503  051525  042103
025240  052040  046511  047505
025246  052125      000
025251      040  050050  024503  DH1:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR'
025256  020040  020040  050050
025264  024523  020040  020040
025272  051450  024520  020040
025300  020040  042524  052123
025306  020040  042040  053105
025314  042101  020122  051040
025322  043505  042101  000122
                        .EVEN
025330  001116  001200  001174  DT1:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,0
025336  001160  001162  001164
025344  000000

                                        ;INFORMATION FOR MESSAGE 2

025346  054523  052123  046505  EM2:    .ASCIZ  'SYSTEM CONTROL REGISTER ERROR'
025354  041440  047117  051124
025362  046117  051040  043505
025370  051511  042524  020122
025376  051105  047522  000122
025404  024040  041520  020051  DM2:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR  WAS     S/B'
025412  020040  024040  051520
025420  020051  020040  024040
025426  050123  020051  020040
025434  052040  051505  020124
025442  020040  042504  040526
025450  051104  020040  042522
025456  040507  051104  020040
025464  053440  051501  020040
025472  020040  051440  041057
025500      000
                        .EVEN
025502  001116  001200  001174  DT2:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4,0
025510  001160  001162  001164
025516  001166  001170  000000

                                        ;INFORMATION FOR MESSAGE 3

025524  044104  030461  046440  EM3:    .ASCIZ  'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'
025532  051501  042524  020122
025540  046103  040505  020122
```

```
                025546  040506  046111  042105
                025554  052040  020117  046103
                025562  020122  050123  041505
                025570  043111  042511  020104
                025576  042522  000107
 5638
 5639                                           ;INFORMATION FOR MESSAGE 4
 5640
 5641   025602  044514  042516  050040  EM4:    .ASCIZ  'LINE PARAMETER REGISTER ERROR'
        025610  051101  046501  052105
        025616  051105  051040  043505
        025624  051511  042524  020122
        025632  051105  047522  000122
 5642
 5643                                           ;INFORMATION FOR MESSAGE 5
 5644
 5645   025640  051102  040505  020113  EM5:    .ASCIZ  'BREAK CONTROL REGISTER ERROR'
        025646  047503  052116  047522
        025654  020114  042522  044507
        025662  052123  051105  042440
        025670  051122  051117     000
 5646
 5647                                           ;INFORMATION FOR MESSAGE 6
 5648
 5649   025675     123  046111  020117  EM6:    .ASCIZ  'SILO STATUS REGISTER ERROR'
        025702  052123  052101  051525
        025710  051040  043505  051511
        025716  042524  020122  051105
        025724  047522  000122
 5650
 5651                                           ;INFORMATION FOR MESSAGE 7
 5652
 5653   025730  052503  051122  047105  EM7:    .ASCIZ  'CURRENT ADDRESS REGISTER ERROR - LINE #XX'
        025736  020124  042101  051104
        025744  051505  020123  042522
        025752  044507  052123  051105
        025760  042440  051122  051117
        025766  026440  046040  047111
        025774  020105  054043  000130
 5654
 5655                                           ;INFORMATION FOR MESSAGE 10
 5656
 5657   026002  054502  042524  041440  EM10:   .ASCIZ  'BYTE COUNTER REGISTER ERROR - LINE #XX'
        026010  052517  052116  051105
        026016  051040  043505  051511
        026024  042524  020122  051105
        026032  047522  020122  020055
        026040  044514  042516  021440
        026046  054130     000
 5658
 5659                                           ;INFORMATION FOR MESSAGE 11
 5660
 5661   026051     125  042516  050130  EM11:   .ASCIZ  'UNEXPECTED DH11 RCVR INTERRUPT'
        026056  041505  042524  020104
```

```
                026064  044104  030461  051040
                026072  053103  020122  047111
                026100  042524  051122  050125
                026106  000124
  5662
  5663                                  ;INFORMATION FOR MESSAGE 12
  5664
  5665  026110  047125  054105  042520  EM12:   .ASCIZ  'UNEXPECTED DH11 XMITTR INTERRUPT'
        026116  052103  042105  042040
        026124  030510  020061  046530
        026132  052111  051124  044440
        026140  052116  051105  052522
        026146  052120     000
  5666
  5667                                  ;INFORMATION FOR MESSAGE 13
  5668
  5669  026151     103  040510  020122  EM13:   .ASCIZ  'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'
        026156  053101  044501  040514
        026164  046102  020105  040506
        026172  046111  042103  052040
        026200  020117  042507  042516
        026206  040522  042524  051040
        026214  053103  020122  047111
        026222  042524  051122  050125
        026230  000124
  5670
  5671                                  ;INFORMATION FOR MESSAGE 14
  5672
  5673  026232  051124  047101  046523  EM14:   .ASCIZ  'TRANSMITTER NPR LOGIC ERROR - LINE #  '
        026240  052111  042524  020122
        026246  050116  020122  047514
        026254  044507  020103  051105
        026262  047522  020122  020055
        026270  044514  042516  021440
        026276  020040     000
  5674
  5675                                  ;INFORMATION FOR MESSAGE 15
  5676
  5677  026301     130  044515  052124  EM15:   .ASCIZ  'XMITTR FAILED TO INTERRUPT - LINE #  '
        026306  020122  040506  046111
        026314  042105  052040  020117
        026322  047111  042524  051122
        026330  050125  020124  020055
        026336  044514  042516  021440
        026344  020040     000
  5678
  5679                                  ;INFORMATION FOR MESSAGE 16
  5680
  5681  026347     122  053103  020122  EM16:   .ASCIZ  'RCVR FAILED TO INTERRUPT'
        026354  040506  046111  042105
        026362  052040  020117  047111
        026370  042524  051122  050125
        026376  000124
  5682
```

```
5683                                        ;INFORMATION FOR MESSAGE 17
5684
5685  026400 051124 047101 046523  EM17:   .ASCIZ  'TRANSMITTER TIMING ERROR - LINE #  '
      026406 052111 042524 020122
      026414 044524 044515 043516
      026422 042440 051122 051117
      026430 026440 046040 047111
      026436 020105 020043 000040
5686  026441 024040 041520 020051  DH6:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  SPEED   TIMEB   TIMEC'
      026447 020040 024040 051520
      026455 020040 024040 051520
      026463 020051 020040 024040
      026471 050123 020051 020040
      026477 052040 051505 020124
      026505 020040 042504 040526
      026513 051104 020040 050123
      026521 042505 020104 020040
      026527 044524 042515 020102
      026535 020040 044524 042515
      026543 000103
5687
5688                                        ;INFORMATION FOR MESSAGE 20
5689
5690  026545 042522 042503 053111  EM20:   .ASCIZ  'RECEIVER TIMING ERROR - LINE #  '
      026553 051105 052040 046511
      026561 047111 020107 051105
      026567 047522 020122 020055
      026575 044514 042516 021440
      026603 020040    000
5691
5692                                        ;INFORMATION FOR MESSAGE 21
5693
5694  026606    122 053103 020122  EM21:   .ASCIZ  'RCVR FAILED TO INTERRUPT - LINE #  '
      026613 040506 046111 042105
      026621 052040 020117 047111
      026627 042524 051122 050125
      026635 020124 020055 044514
      026643 042516 021440 020040
      026651    000
5695
5696                                        ;INFORMATION FOR MESSAGE 22
5697
5698  026652    103 040510 020122  EM22:   .ASCIZ  'CHAR AVAIL FAILED TO SET ON TIME - LINE #  '
      026657 053101 044501 020114
      026665 040506 046111 042105
      026673 052040 020117 042523
      026701 020124 047117 052040
      026707 046511 020105 020055
      026715 044514 042516 021440
      026723 020040    000
5699
5700                                        ;INFORMATION FOR MESSAGE 23
5701
5702  026726    102 051501 041511  EM23:   .ASCIZ  'BASIC DATA TEST ERROR - LINE #  '
      026733 042040 052101 020101
```

```
                026736  042524  052123  042440
                026744  051122  051117  026440
                026752  046040  047111  020105
                026760  020043  000040
 5703           026764  024040  041520  020051  DH7:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG   WAS     S/B'
                026772  020040  024040  051520
                027000  020051  020040  024040
                027006  050123  020051  020040
                027014  052040  051505  020124
                027022  020040  042504  040526
                027030  051104  020040  044103
                027036  046122  043516  020040
                027044  053440  051501  020040
                027052  020040  051440  041057
                027060     000
 5704
 5705                                   ;INFORMATION FOR MESSAGE 24
 5706
 5707           027061     101  052125  020117  EM24:   .ASCIZ  'AUTO ECHO TEST ERROR - LINE #  '
                027066  041505  047510  052040
                027074  051505  020124  051105
                027102  047522  020122  020055
                027110  044514  042516  021440
                027116  020040     000
 5708
 5709                                   ;INFORMATION FOR MESSAGE 25
 5710
 5711           027121     102  042522  045501  EM25:   .ASCIZ  'BREAK BIT TEST ERROR - LINE #  '
                027126  041040  052111  052040
                027134  051505  020124  051105
                027142  047522  020122  020055
                027150  044514  042516  021440
                027156  020040     000
 5712
 5713                                   ;INFORMATION FOR MESSAGE 26
 5714
 5715           027161     110  046101  026506  EM26:   .ASCIZ  'HALF-DUPLEX TEST ERROR - LINE #  '
                027166  052504  046120  054105
                027174  052040  051505  020124
                027202  051105  047522  020122
                027210  020055  044514  042516
                027216  021440  020040     000
 5716
 5717                                   ;INFORMATION FOR MESSAGE 27
 5718
 5719           027223     125  042516  050130  EM27:   .ASCIZ  'UNEXPECTED BUS ERROR TRAP'
                027230  041505  042524  020104
                027236  052502  020123  051105
                027244  047522  020122  051124
                027252  050101     000
 5720           027255     040  050050  024503  DH3:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   TRPPC   TRPPS '
                027262  020040  020040  050050
                027270  024523  020040  020040
                027276  051450  024520  020040
```

```
        027304  020040  042524  052123
        027312  020040  052040  050122
        027320  041520  020040  052040
        027326  050122  051520  000040
5721                                    .EVEN
5722    027334  001116  001200  001174  DT3:    .WORD   $ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,0
        027342  001160  001162  001164
        027350  000000
5723
5724                                    ;INFORMATION    FOR MESSAGE 30
5725
5726    027352  047125  054105  042520  EM30:   .ASCIZ  'UNEXPECTED RSVD INSTR TRAP'
        027360  052103  042105  051040
        027366  053123  020104  047111
        027374  052123  020122  051124
        027402  050101     000
5727
5728                                    ;INFORMATION FOR MESSAGE 31
5729
5730    027405     101  052125  020117  EM31:   .ASCIZ  'AUTO ECHO DATA COMPARE ERROR - LINE #  '
        027412  041505  047510  042040
        027420  052101  020101  047503
        027426  050115  051101  020105
        027434  051105  047522  020122
        027442  020055  044514  042516
        027450  021440  020040     000
5731    027455     040  050050  024503  DH4:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST    WASADR  SBADR    WAS     S/B'
        027463  020040  020040  050050
        027470  024523  020040  020040
        027476  051450  024520  020040
        027504  020040  042524  052123
        027512  020040  053440  051501
        027520  042101  020122  051440
        027526  040502  051104  020040
        027534  020040  040527  020123
        027542  020040  020040  027523
        027550  000102
5732
5733                                    ;INFORMATION FOR MESSAGE 32
5734
5735    027552  052501  047524  042440  EM32:   .ASCIZ  'AUTO ECHO TEST TIMEOUT - LINE #  '
        027560  044103  020117  042524
        027566  052123  052040  046511
        027574  047505  052125  026440
        027602  046040  047111  020105
        027610  020043  000040
5736    027614  024040  041520  020051  DH5:    .ASCIZ  ' (PC)   (LPRG)   TEST'
        027622  020040  046050  051120
        027630  024507  020040  052040
        027636  051505  000124
5737                                    .EVEN
5738    027642  001116  001200  001204  DT4:    .WORD   $ERRPC,$TMP0,$TMP2,0
        027650  000000
5739
```

```
                                        ;INFORMATION FOR MESSAGE 33
                                        
027652 040520 044522 054524 EM33:   .ASCIZ  'PARITY LOGIC TEST ERROR - LINE #  '
027660 046040 043517 041511
027666 052040 051505 020124
027674 051105 047522 020122
027702 020055 044514 042516
027710 021440 020040    000
                                        
                                        ;INFORMATION FOR MESSAGE 34
                                        
027715    115 046125 044524 EM34:   .ASCIZ  'MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  '
027722 046055 047111 020105
027730 040520 044522 054524
027736 042040 052101 020101
027744 042524 052123 042440
027752 051122 051117 026440
027760 046040 047111 020105
027766 020043 020040 020055
027774 052523 052122 051505
030002 020124 020043 000040
                                        
                                        ;INFORMATION FOR MESSAGE 35
                                        
030010 052515 052114 026511 EM35:   .ASCIZ  'MULTI-LINE PARITY DATA TEST TIMEOUT'
030016 044514 042516 050040
030024 051101 052111 020131
030032 040504 040524 052040
030040 051505 020124 044524
030046 042515 052517 000124
030054 024040 041520 020051 DH14:   .ASCIZ  ' (PC)   (LPRG)  ACTLIN'
030062 020040 046050 051120
030070 024507 020040 041501
030076 046124 047111    000
030104                      .EVEN
030104 001116 001200 001206 DT6:    .WORD   SERRPC,STMPO,STMP3,0
030112 000000
                                        
                                        ;INFORMATION FOR MESSAGE 36
                                        
030114 044103 051101 040440 EM36:   .ASCIZ  'CHAR AVAILABLE TIMEOUT'
030122 040526 046111 041101
030130 042514 052040 046511
030136 047505 052125    000
                                        
                                        ;INFORMATION FOR MESSAGE 37
                                        
030143    104 052101 020101 EM37:   .ASCIZ  'DATA COMPARE ERROR - LINE #  '
030150 047503 050115 051101
030156 020105 051105 047522
030164 020122 020055 044514
030172 042516 021440 020040
030200    000
```

```
5763                                            ;INFORMATION FOR MESSAGE 40
5764
5765  030201    102  043125  042506  EM40:   .ASCIZ  'BUFFER ACTIVE REG ERROR - LINE #  '
      030206 020122  041501  044524
      030214 042526  051040  043505
      030222 042440  051122  051117
      030230 026440  046040  047111
      030236 020105  020043  000040
5766
5767                                            ;INFORMATION FOR MESSAGE 41
5768
5769  030244 041522  051126  043040  EM41:   .ASCIZ  'RCVR FALSE INTERRUPT'
      030252 046101  042523  044440
      030260 052116  051105  052522
      030266 052120     000
5770
5771                                            ;INFORMATION FOR MESSAGE 42
5772
5773  030271    123  046111  020117  EM42:   .ASCIZ  'SILO OVERFLOW ERROR'
      030276 053117  051105  046106
      030304 053517  042440  051122
      030312 051117     000
5774
5775                                            ;INFORMATION FOR MESSAGE 43
5776
5777  030315    123  046111  020117  EM43:   .ASCIZ  'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
      030322 053117  051105  046106
      030330 053517  043040  044501
      030336 042514  020104  047524
      030344 043440  047105  051105
      030352 052101  020105  041522
      030360 051126  044440  052116
      030366 051105  052522  052120
      030374    000
5778
5779                                            ;INFORMATION FOR MESSAGE 44
5780
5781  030375    116  047117  042440  EM44:   .ASCIZ  'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
      030402 020130  042515  047515
      030410 054522  043040  044501
      030416 042514  020104  047524
      030424 043440  047105  051105
      030432 052101  020105  046530
      030440 052111  051124  044440
      030446 052116  051105  052522
      030454 052120     000
5782
5783                                            ;INFORMATION FOR MESSAGE 45
5784
5785  030457    130  044515  020124  EM45:   .ASCIZ  'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
      030464 047504  042516  043040
      030472 044501  042514  020104
      030500 047524  043440  047105
      030506 051105  052101  020105
```

```
                030514  046530  052111  051124
                030522  044440  052116  051105
                030530  052522  052120     000
 5786
 5787                                           ;INFORMATION FOR MESSAGE 46
 5788
 5789   030535     103  051125  042522  EM46:   .ASCIZ  'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  '
        030542  052116  042440  042104
        030550  042522  051523  046440
        030556  046505  051117  020131
        030564  040520  052124  051105
        030572  051516  052040  051505
        030600  020124  051105  047522
        030606  020122  020055  044514
        030614  042516  021440  020040
        030622     000
 5790   030623     040  050050  024503  DH10:   .ASCIZ  ' (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B'
        030630  020040  046540  047111
        030636  053505  020122  050040
        030644  052101  051124  020116
        030652  020040  042524  052123
        030660  020040  042040  053105
        030666  042101  020122  051040
        030674  043505  042101  020122
        030702  020040  040527  020123
        030710  020040  020040  027523
        030716  000102
 5791                                           .EVEN
 5792   030720  001116  001200  001202  DTS:    .WORD   SERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4,0
        030726  001160  001162  001164
        030734  001166  001170  000000
 5793
 5794                                           ;INFORMATION FOR MESSAGE 47
 5795
 5796   030742  054502  042524  041440  EM47:   .ASCIZ  'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  '
        030750  052517  052116  046440
        030756  046505  051117  020131
        030764  040520  052124  051105
        030772  051516  052040  051505
        031000  020124  051105  047522
        031006  020122  020055  044514
        031014  042516  021440  020040
        031022     000
 5797                                           ;INFORMATION FOR MESSAGE 50
 5798
 5799   031023     124  051505  020124  EM50:   .ASCIZ  'TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
        031030  044524  042515  052517
        031036  020124  040527  052111
        031044  047111  020107  047506
        031052  020122  046530  052111
        031060  042040  047117  020105
        031066  020055  044514  042516
        031074  021440  020040     000
 5800
```

```
                                        ;INFORMATION FOR MESSAGE 51
                                        ;
031101      116  051120  046040  EMS1:   .ASCIZ  'NPR LOGIC TEST 2 ERROR'
031106  043517  041511  052040
031114  051505  020124  020062
031122  051105  047522  000122
031130  024040  041520  020051  DH11:   .ASCIZ  ' (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B'
031136  020040  044514  040516
031144  052103  020040  044514
031152  041516  045510  020040
031160  052040  051505  020124
031166  020040  042504  040526
031174  051104  020040  042522
031202  040507  051104  020040
031210  053440  051501  020040
031216  020040  051440  041057
031224      000
                                        ;INFORMATION FOR MESSAGE 52
                                        ;
031225      102  051501  041511  EMS2:   .ASCIZ  'BASIC DATA COMPARE ERROR'
031232  042040  052101  020101
031240  047503  050115  051101
031246  020105  051105  047522
031254  000122
                                        ;INFORMATION FOR MESSAGE 53
                                        ;
031256  024040  041520  020051  DH12:   .ASCIZ  ' (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B'
031264  020040  050123  042505
031272  020104  020040  024040
031300  050123  020051  020040
031306  052040  051505  020124
031314  020040  042504  040526
031322  051104  020040  042522
031330  040507  051104  020040
031336  053440  051501  020040
031344  020040  051440  041057
031352      000
                                        ;INFORMATION FOR MESSAGE 55
                                        ;
031353      040  050050  024503  DH13:   .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B'
031360  020040  020040  050050
031366  024523  020040  020040
031374  051450  024520  020040
031402  020040  042524  052123
031410  020040  042040  053105
031416  042101  020122  041440
031424  051110  047114  020107
031432  051440  051103  040527
031440  020123  051440  051103
031446  027523  000102
                                        ;
```

```
                                                ;INFORMATIOPN FOR MESSAGE 56
                                                ;
031452 053117 051105 052522  EM56:   .ASCIZ  'OVERRUN BIT FAILED TO SET - LINE #  '
031460 020116 044502 020124
031466 040506 046111 042105
031474 052040 020117 042523
031502 020124 020055 044514
031510 042516 021440 020040
031516    000
                                                ;
                                                ;INFORMATION FOR MESSAGE 57
                                                ;
031517    123 047524 040522  EM57:   .ASCIZ  'STORAGE OVERFLOW BIT FAILED - LINE #  '
031524 042507 047440 042526
031532 043122 047514 020127
031540 044502 020124 040506
031546 046111 042105 026440
031554 046040 047111 020105
031562 020043 000040
                                                .EVEN
                                                ;MISCELLANEOUS MESSAGES
                                                ;
031566 005015 040515 047111  TITLE:  .ASCIZ  <15><12>'MAINDEC-11-DZDHM-A  DH11 DIAGNOSTIC'<15><12>
031574 042504 026503 030461
031602 042055 042132 046510
031610 040455 020040 044104
031616 030461 042040 040511
031624 047107 051517 044524
031632 006503 000012
031636 005015 042524 052123  TITLE2: .ASCIZ  <15><12>'TESTING DH11 #  '<15><12>
031644 047111 020107 044104
031652 030461 021440 020040
031660 005015    000
031663    015 052012 050131  INMSG1: .ASCIZ  <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>
031670 020105 041523 020122
031676 042101 051104 051505
031704 020123 047506 020122
031712 044506 051522 020124
031720 044104 030461 005015
031726    000
031727    015 052012 050131  INMSG2: .ASCIZ  <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
031734 020105 042526 052103
031742 051117 040440 042104
031750 042522 051523 043040
031756 051117 043040 051111
031764 052123 042040 030510
031772 006461 000012
032000 005015 054524 042520  INMSG3: .ASCIZ  <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
032004 042040 030510 020061
032012 042504 044524 042503
032020 051440 046105 041505
032026 044524 047117 050040
032034 051101 046501 052105
032042 051105 005015    000
```

```
5833  032047  000015  044412  053116  INMSG4: .ASCIZ  <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
      032054  046101  042111  042040
      032062  030510  020061  041523
      032070  020122  042101  051104
      032076  051505  020123  020055
      032104  051124  020131  043501
      032112  044501  006516  000012
5834  032120  005015  047111  040526  INMSG5: .ASCIZ  <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
      032126  044514  020104  044104
      032134  030461  053040  041505
      032142  047524  020122  042101
      032150  051104  051505  020123
      032156  020055  051124  020131
      032164  043501  044501  006516
      032172  000012
5835  032174  005015  054524  042520  INMSG6: .ASCIZ  <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
      032202  046040  047111  020105
      032210  042523  042514  052103
      032216  047511  020116  040520
      032224  040522  042515  042524
      032232  006522  000012
5836  032236  005015  042504  051120  INMSG7: .ASCIZ  <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
      032244  051505  020123  041442
      032252  047117  044524  052516
      032260  021105  052040  020117
      032266  052123  051101  020124
      032274  042524  052123  047111
      032302  006507  000012
5837
5838  032306  005015  054524  042520  VCMC:   .ASCIZ  <15><12>'TYPE NO. OF WORDS (OCTAL) BETWEEN VECTORS (4 OR 10)'<15><12>
      032314  047040  027117  047440
      032322  020106  047527  042122
      032330  020123  047450  052103
      032336  046101  020051  042502
      032344  053524  042505  020116
      032352  042526  052103  051117
      032360  020123  032050  047440
      032366  020122  030061  006451
      032374  000012
5839                          .EVEN
5840                          ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3
5841
5842  032376  000020          RCNT:   .BLKW   16.
5843
5844                          ;256. WORD RECEIVER INPUT BUFFER
5845
5846  032436  000400          RBUF:   .BLKW   256.
5847
5848
5849                          ;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
5850
5851                          .EVEN
5852  033436  000400          TBUF:   .BLKB   256.
5853
```

```
  5854          000001                  .END
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
ABASE = 000000          2259
ACDMI = 000000          2259
ACDM2 = 000000          2259
ACPTOP= 000000          2259
ADDMD = 000000          2259
ADDMI = 000000          2259
ADDMIO= 000000          2259
ADDMII= 000000          2259
ADDMI2= 000000          2259
ADDMI3= 000000          2259
ADDMI4= 000000          2259
ADDMI5= 000000          2259
ADDM2 = 000000          2259
ADDM3 = 000000          2259
ADDM4 = 000000          2259
ADDM5 = 000000          2259
ADDM6 = 000000          2259
ADDM7 = 000000          2259
ADDM8 = 000000          2259
ADDM9 = 000000          2259
ADEVCT= 000000          2259
ADEVM = 000000          2259
ADPTR   025150          2619#   2626#   2634    4975#   5599#
AENV  = 000000          2259
AENVM = 000000          2259
AETABL  024516          4520    4597    4694    4728    4732    4733    5435#
AETABO  024556          4589    5455#
AFATAL= 000000          2259
AMADRI= 000000          2259
AMADR2= 000000          2259
AMADR3= 000000          2259
AMADR4= 000000          2259
AMAMSI= 000000          2259
AMAMS2= 000000          2259
AMAMS3= 000000          2259
AMAMS4= 000000          2259
AMSGMO= 000000          2259
AMSGLG= 000000          2259
AMSGTY= 000000          2259
AMTYPI= 000000          2259
AMTYP2= 000000          2259
AMTYP3= 000000          2259
AMTYP4= 000000          2259
APASS = 000000          2259
APRIOR= 000000          2259
APTCSU= 000040          4990    4991#
APTENV= 000001          4986    4990    4991#
APTSIZ= 000200          2596    4991#
APTSPO= 000100          4990    4991#
ASWREG= 000000          2259
ATESTM= 000000          2259
AUNIT = 000000          2259
AUSWR = 000000          2259
AVECTI= 000000          2259
```

```
AVECT2= 000000  2259
BAR   = 000012  3644* 3675  3679  3735* 3996* 4065* 4130* 4196* 4253* 4306  4368* 4429* 4490
                4496  4540* 4622* 4707* 4774* 4806* 4810  4819  4879* 4883  4890  4892  4927*
                5383*
BCR   = 000010  3075  3155  3235  3326  3642* 3698  3703  3760  3771  3994* 4062* 4127* 4192*
                4249* 4320  4365* 4534* 4617* 4697* 4772* 4803* 4877* 4925* 5120* 5132* 5147*
                5329* 5382*
BEGIN   002144  2248  2593*
BEGINA  002146  2594* 5185  5188
BIT0    = 000001  2258*
BIT00   = 000001  2258* 5475
BIT01   = 000002  2258* 5476
BIT02   = 000004  2258* 5477
BIT03   = 000010  2258* 5478
BIT04   = 000020  2258* 5479
BIT05   = 000040  2258* 5480
BIT06   = 000100  2258* 3478  3807  5481
BIT07   = 000200  2258* 3479  3657  4081  4147  5482
BIT08   = 000400  2258* 4985  5483
BIT09   = 001000  2258* 2767  2769  2780  2781  2792  3477  3517  3557  3597  4252  4985  4986
                  5484
BIT1    = 000002  2258*
BIT10   = 002000  2258* 3559  5485
BIT11   = 004000  2258* 2688  3439  3474  3514  3554  3594  3640  3709  3730  3805  3858  3891
                  3935  3990  4060  4125  4190  4246  4358  4419  4445  4452  4466  4497  4511
                  4519  4611  4692  4767  4800  4873  4922  4985  5486
BIT12   = 010000  2258* 3519  5487
BIT13   = 020000  2258* 3558  3598  3645  4986  5488
BIT14   = 040000  2258* 3518  4000  4006  4022  4449  4953  4985  5489
BIT15   = 100000  2258* 3599  3747  3806  3861  3865  3896  3899  3939  3943  4083  4147  4554
                  4583  4682  4745  4907  4940  4953  5490
BIT2    = 000004  2258*
BIT3    = 000010  2258*
BIT4    = 000020  2258*
BIT5    = 000040  2258*
BIT6    = 000100  2258* 3981
BIT7    = 000200  2258*
BIT8    = 000400  2258*
BIT9    = 001000  2258*
BKR     = 000014  2679  2843  2898* 2952  3014* 3397* 4367* 4805* 5384*
BPTVEC= 000014  2258*
BRKTAB  024720  4753  5504*
BRLVL   025100  2621  5576*
BRPTR   025154  2621* 2628* 2636  4977* 5601*
BUSER   023774  2599  5269*
CAR   = 000006  3029  3121  3189  3281  3381* 3391* 3398  3686  3692  3776  3787  3993* 4063*
                4128* 4193* 4248* 4334  4366* 4535* 4536* 4615* 4616* 4696* 4771* 4802* 4876*
                4924* 5119* 5146* 5328* 5381*
CHKACR  023632  5195  5229*
CHKVCT  023714  5203  5248*
CHPS1   024304  3442  3480  3520  3560  3600  3646  3715  3808  3844  4432  4436  5279  5292
                5347*
CHPS2   024320  3438  3476  3516  3556  3596  3639  3803  4423  5355*
CLCABC  023176  3638  3731  4532  4768  4874  5117*
```

```
DDISP = 177570          2258#  2259
DHAOR   024400          2634*  4245    4304    4415    4610    4985    5084*  5387#
DHAOTB  024776          2619   5238    5241    5532#
DHNUM   025142          2622*  2625*   2638    4974*   5594#
DHRLVL  025140          2636*  3426    3470    3510    3550    3590    3802   4426   5592#
DHSEL   024406          2629   4980    5211*   5390#
DHTLVL  025141          3429   3473    3513    3553    3593    3630    5593#
DHVCT   024402          2635*  3424    3468    3508    3548    3588    3627   3710   3800   3846   4424   5298   5388#
DHVCTB  025036          2620   5257    5260    5552#
DH1     025251          2623   5624#
DH10    030623          2524   2531    5790#
DH11    031130          2545   5804#
DH12    031256          2559   2566    5812#
DH13    031353          2573   5816#
DH14    030054          2461   5751#
DH2     025404          2270   2277    2284    2291    2298    2305    2312   2319   2326   2333   2340   2347   2354
                        2375   2382    2397    2404    2411    2447    2482   2503   2510   2517   2538   2552   2580
                        2587   5631#
DH3     027255          2418   2425    5720#
DH4     027455          2432   2454    2475    5731#
DH5     027614          2439   2468    2489    2496    5736#
DH6     026444          2361   2368    5686#
DH7     026764          2389   5703#
DISPLA  001140          2259#  2596*   4985*   4986*
DISPRE  000174          2247#  2596
DSWR = 177570           2258#  2259
DT1     025330          2264   5626#
DT2     025502          2271   2278    2285    2292    2299    2306    2313   2320   2327   2334   2341   2348   2355
                        2362   2369    2376    2383    2390    2398    2405   2412   2433   2448   2455   2476   2483
                        2504   2511    2518    2539    2553    2560    2567   2574   2581   2588   5633#
DT3     027334          2419   2426    5722#
DT4     027642          2440   2469    2490    2497    5738#
DT5     030720          2525   2532    2546    5792#
DT6     030104          2462   5753#
EM1VEC= 000030          2258#  2596*
EM1     025202          2262   5623#
EM10    026002          2311   3105    3162    4330    5657#
EM11    026051          2318   5661#
EM12    026110          2325   5665#
EM13    026151          2332   5669#
EM14    026232          2339   3721    5673#
EM15    026301          2346   3661    5677#
EM16    026347          2353   5681#
EM17    026400          2360   4107    5685#
EM2     025346          2269   5630#
EM20    026542          2367   4171    5690#
EM21    026603          2374   5694#
EM22    026647          2381   2565    2572    4152    4214    4272    4386   5698#
EM23    026723          2388   4229    5702#
EM24    027061          2396   4560    4576    5707#
EM25    027121          2403   4795    4825    4844    4861    5711#
EM26    027161          2410   4897    4912    5715#
EM27    027223          2417   5719#
EM3     025524          2276   5637#
```

```
EM30    027352          2424    5726#
EM31    027405          2431    4658    4673    4738    5730#
EM32    027552          2438    4635    5735#
EM33    027652          2446    4400    5742#
EM34    027715          2453    4473    4476    5746#
EM35    030010          2460    5750#
EM36    030114          2467    5757#
EM37    030143          2474    4295    5761#
EM4     02[illegible]02 2283    5641#
EM40    02[illegible]01 2481    4316    5765#
EM41    02[illegible]44 2488    5769#
EM42    02[illegible]21 2495    5773#
        [illegible]     2502    5777#
        [illegible]     2509    5781#
        [illegible]     2516    5785#
EM6     0[illegible]5   2523    3222    3313    5789#
EM7     03[illegible]42 2530    3268    3358    5796#
EM5     02[illegible]40 2290    5645#
EM0     031023          2537    [illegible]58   3753    4088    4945    5799#
EM1     031101          2544    [illegible]3#
EM2     031225          2551    5[illegible]8#
EM6     031452          2579    4966    5820#
EM57    031517          2586    4013    4033    4047    5824#
EM6     02[illegible]75 2247    5649#
EM7     025730          2304    3059    3128    3410    4344    5653#
ENDA    017674          4921    4972#   4981
ERRVEC= 000004          2258#   2599*   2600*   2652    2653*   2668*   4985*
EXPAR   023626          5222    5226#
GNS     = ****** U      2247    4994
INMSG1  031663          5191    5830#
IN[illegible]G2 031727  5199    5831#
IN[illegible]23 031776  5207    5832#
IN[illegible]47 032047  5245    5833#
IN[illegible]G5 032120  5264    5834#
IN[illegible]G6 032174  5214    5835#
IN[illegible]G  03[illegible]6  5224  5836#
IN[illegible]   02[illegible]   2617    5190#   5196
IN[illegible]   02[illegible]   2610    5167#   5176
IN[illegible]   02[illegible]   2249    5184#
IN[illegible]   02[illegible]   2247    5187#
I[illegible]    0[illegible]    5[illegible]4   5[illegible]#   5204
I[illegible]    0[illegible]    2[illegible]16  [illegible]2    5206#
I[illegible]    0[illegible]    5[illegible]10  5[illegible]    #
IN[illegible]   0[illegible]    3431    3[illegible]7#
IO[illegible]   =[illegible]    2[illegible]8*  2[illegible]6#
[illegible]     [illegible]     3732    5130#
[illegible]     [illegible]     2642    5005#
LIN[illegible]  0[illegible]56  4428*   4483*   4504    4510    4621*   4679*   4680    4706*   4746*   5339*   5492#
LIN[illegible]  024616          4483    4679    4746    5475#
LINE    025144          3034    3038    3046    3050    3058    3066*   3080    3084    3092    3096    3104    3112*   3125
                        3127    3159    3161    3195    3213    3241    3259    3286    3304    3331    3349    3388    3401
                        3637    3641    3660    3720    3733    3748    3752    3761    3768    3777    3784    3992    4007
                        4012    4027    4032    4046    4061    4084    4087    4106    4126    4148    4151    4170    4191
                        4210    4213    4220    4228    4247    4268    4271    4285    4294    4315    4329    4343    4360
```

```
                    4362   4381   4385   4399   4533   4556   4559   4575   4588*  4614   4634   4644   4672
                    4769   4794   4801   4824   4843   4860   4875   4896   4908   4911   4923   4939   4944
                    4950   4965   4985*  5070   5072*  5074*  5076*  5083*  5595#
LINEA   025146      3195*  3204   3241*  3250   3286*  3295   3331*  3340   5051   5596#
LINMSK  024412      3644   3735   3996   4065   4130   4196   4253   4367   4368   4540   4621   4774   4905
                    4806   4879   4927   5073*  5077*  5079   5392#
LINSEL  024410      3030   3031*  3069*  3076   3077*  3115*  4422   4428   4429   4528   4605   4622   4706
                    4707   4761   5079   5218*  5220*  5339   5391#
LMSK1   024414      4521*  4525*  4528   4598*  4602*  4605   4754*  4758*  4761   5393#
LPR     = 000004    2680   2794*  2809   2887*  2914   3002*  3390*  3643*  3734*  3995*  4064*  4129*  4194*
                    4195*  4250*  4364*  4421   4539*  4551*  4570*  4583*  4618*  4632   4682*  4698*  4717
                    4745*  4773*  4804*  4878*  4926*  5148*  5330*  5380#
MSTCLR  024416      2673   2685   5395#
MULPTB  024660      4481*  5157   5159   5499#
NRC     = 000002    3821   3827   3905   4017   4018   4218   4276   4390   4396   4456   4468   4565   4572
                    4626   4721   4734   4850   4857   4954   4962   5379#
PATRNA  024426      3190   3236   5398#
PATRNB  024436      3372   5403#
PC      =%000007    2258#  2610*  2642*  2662*  2692*  2718*  2727*  2749*  2776*  2788*  2800*  2825*  2834*
                    2859*  2868*  2894*  2905*  2932*  2943*  2970*  2981*  3009*  3021*  3032*  3044*  3056*
                    3078*  3090*  3102*  3122*  3136*  3146*  3156*  3170*  3180*  3193*  3219*  3239*  3265*
                    3284*  3310*  3329*  3355*  3386*  3407*  3438*  3442*  3446*  3448*  3453*  3455*  3461*
                    3476*  3480*  3483*  3488*  3492*  3497*  3501*  3516*  3520*  3523*  3528*  3532*  3537*
                    3541*  3556*  3560*  3563*  3568*  3572*  3577*  3581*  3596*  3600*  3603*  3608*  3612*
                    3617*  3621*  3631*  3638*  3639*  3646*  3651*  3655*  3658*  3668*  3670*  3671*  3678*
                    3681*  3682*  3690*  3693*  3694*  3701*  3704*  3705*  3715*  3722*  3728*  3731*  3732*
                    3741*  3745*  3750*  3772*  3788*  3803*  3809*  3813*  3816*  3826*  3828*  3838*  3840*
                    3844*  3876*  3918*  3950*  3951*  3952*  3959*  3962*  3963*  3973*  3988*  4002*  4005*
                    4010*  4025*  4030*  4040*  4044*  4056*  4071*  4085*  4098*  4104*  4121*  4136*  4149*
                    4162*  4168*  4185*  4202*  4206*  4211*  4226*  4241*  4259*  4263*  4269*  4292*  4310*
                    4313*  4324*  4327*  4338*  4341*  4356*  4374*  4378*  4383*  4394*  4397*  4420*  4423*
                    4432*  4436*  4437*  4465*  4470*  4492*  4506*  4530*  4532*  4546*  4550*  4557*  4569*
                    4573*  4607*  4612*  4628*  4650*  4655*  4666*  4670*  4713*  4731*  4735*  4763*  4768*
                    4780*  4794*  4792*  4812*  4816*  4822*  4833*  4841*  4855*  4858*  4871*  4874*  4885*
                    4889*  4894*  4904*  4909*  4920*  4933*  4936*  4942*  4960*  4963*  4984*  4986*  4987*
                    4990*  4991*  4995   5011*  5016*  5023*  5028*  5037*  5045*  5058*  5085*  5125*  5137*
                    5162*  5180*  5195*  5203*  5246*  5265*  5275*  5279*  5288*  5292*  5305*  5318*  5341*
                    5350*  5358*  5372*
PIRQ    = 177772    2258#
PIRQVE= 000240      2258#
PRTYTB  024456      4408   4412   5415#
PR0     = 000000    2258#
PR1     = 000040    2258#
PR2     = 000100    2258#
PR3     = 000140    2258#
PR4     = 000200    2258#
PR5     = 000240    2258#
PR6     = 000300    2258#
PR7     = 000340    2258#
PS      = 177776    2258#
PSW     = 177776    2258#
PWRVEC= 000024      2258#  2596*  4995*
RBUF    032436      4251   4277   4281   4301   4462   4467   4469   4480*  4647   4652   4653   4677*  5149*
                    5150*  5331*  5332*  5846#
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
RCNT    032376          4695    4727*   4742    5842#
RDCHR = 104405          4992    4994#
RDLIN = 104406          4993    4994#
RDOCT = 104407          4994#   5169    5192    5200    5208    5215
RESERR  024050          2601    5282#
RESTRP  024124          3461    3501    3541    3581    3621    4437    5298#
RESTRT  002544          2625#   2633
RESTI   002602          2631#   5280    5293
RESVEC= 000010          2258#   2601*   2602*
RGMSK1  024760          2704    5521#
RGMSK2  024762          2736    5522#
RGMSK3  024764          2811    2916    2922    2926    5523#
RGMSK4  024766          2845    2954    2960    2963    5524#
RGMSK5  024770          2879    2992    2998    3001    5525#
RGMSK6  024772          2760    5526#
RSTRTA  002612          2630    2634#   4983    4995
R0      =%000000        2258#   2593*   2612    2614    4984*   4987*   4989*   4990*   4991*   4993*   4994*   4995*   5017*
                        5018    5029*   5030    5042    5052*   5053    5097    5098*   5100*   5101    5108*   5109*   5110
                        5111*   5170*   5172    5174    5177*   5179*   5184*   5259    5273*   5286*
R1      =%000001        2258#   2649    2674    2675    2676    2677    2688*   2701    2712*   2722*   2734    2744*   2758
                        2770*   2782*   2794*   2808    2819*   2829*   2842    2853*   2863*   2876    2887*   2898*   2913
                        2924*   2937*   2951    2964*   2975*   2989    3002*   3014*   3028    3038*   3039*   3050*   3051*
                        3074    3084*   3085*   3096*   3097*   3120    3125*   3130*   3141*   3154    3159*   3164*   3175*
                        3188    3198*   3204*   3205*   3210*   3211*   3234    3244*   3250*   3251*   3256*   3257*   3280
                        3289*   3295*   3296*   3301*   3302*   3325    3334*   3340*   3341*   3346*   3347*   3370    3378*
                        3379*   3380*   3381*   3388*   3389*   3390*   3391*   3395*   3396*   3397*   3398    3423    3439*
                        3441*   3447    3454    3467    3474*   3477*   3478*   3479*   3484    3485*   3486*   3493    3495*
                        3496*   3507    3514*   3517*   3518*   3519*   3524    3525*   3526*   3533    3535*   3536*   3547
                        3554*   3557*   3558*   3559*   3564    3565*   3566*   3573    3575*   3576*   3587    3594*   3597*
                        3598*   3599*   3604    3605*   3606*   3613    3615*   3616*   3635    3640*   3641*   3642*   3643*
                        3644*   3645*   3656    3665    3669    3675    3686    3698    3709*   3730*   3733*   3734*   3735*
                        3739    3746    3749    3758*   3760    3770    3776    3786    3805*   3806*   3807*   3814    3820
                        3821    3845*   3855    3858*   3868    3891*   3905    3932    3935*   3945    3990*   3991    3992*
                        3993*   3994*   3995*   3996*   4000    4008    4017    4018    4022    4028    4037    4043    4060*
                        4061*   4062*   4063*   4064*   4065*   4069    4080    4082    4125*   4126*   4127*   4128*   4129*
                        4130*   4134    4145    4146    4190*   4191*   4192*   4193*   4194*   4195*   4196*   4200    4207
                        4218    4245*   4246*   4247*   4248*   4249*   4250*   4252*   4253*   4257    4264    4276    4281*
                        4288    4300*   4301    4304*   4305    4319    4333    4358*   4362*   4364*   4365*   4366*   4367*
                        4368*   4372    4379    4382    4390    4395    4415*   4419*   4421    4427*   4429*   4442    4445*
                        4449    4452*   4456    4466*   4468*   4490    4496    4497*   4511*   4519*   4533*   4534*   4535*
                        4536*   4539*   4540*   4544    4551*   4552    4553    4565    4570*   4571    4583*   4610*   4611*
                        4614*   4615*   4616*   4617*   4618*   4622*   4626    4632    4654*   4669*   4682*   4683    4692*
                        4696*   4697*   4698*   4701*   4707*   4711    4717    4721    4734*   4744*   4745*   4767*   4769*
                        4771*   4772*   4773*   4774*   4778    4786    4800*   4801*   4802*   4803*   4804*   4805*   4806*
                        4810    4818    4830    4835    4850    4856    4873*   4875*   4876*   4877*   4878*   4879*   4883
                        4890    4891    4901    4905    4906    4922*   4923*   4924*   4925*   4926*   4927*   4931    4937
                        4941    4954    4961    4985*   4989*   4991*   4993*   4995*   5005*   5007*   5019    5031    5043
                        5054    5097    5099*   5107*   5110*   5111*   5118*   5119*   5120*   5124*   5131*   5132*   5136*
                        5142*   5146*   5147*   5148*   5151*   5161*   5193*   5201*   5209*   5211    5216*   5218    5229
                        5232    5235    5239    5240*   5248    5251    5254    5258    5259*   5271*   5284*   5327*   5328*
                        5329*   5330*   5333*   5340*
R2      =%000002        2258#   2649*   2650    2654*   2656*   2657    2658    2684*   2687*   2689    2701*   2711*   2713*
                        2714    2723*   2724    2734*   2743*   2745*   2746    2758*   2768*   2769*   2771*   2772    2780*
                        2783*   2784    2793*   2795*   2796    2808*   2809*   2818*   2820*   2821    2830*   2831    2842*
```

```
                    2843*  2852*  2854*  2855   2864*  2865   2876*  2877*  2886*  2888*  2889   2899*  2900
                    2913*  2914*  2923*  2926*  2927*  2928   2938*  2939   2951*  2952*  2962*  2963*  2965*
                    2966   2976*  2977   2989*  2990*  3000*  3001*  3003*  3004   3015*  3016   3028*  3029*
                    3040*  3052   3074*  3075*  3086*  3098   3120*  3121*  3131*  3132   3142*  3143   3154*
                    3155*  3165*  3166   3176*  3177   3188*  3189*  3199*  3206*  3212   3234*  3235*  3245*
                    3252*  3258   3280*  3281*  3290*  3297*  3303   3325*  3326*  3335*  3342*  3348   3370*
                    3371*  3399   3423*  3467*  3507*  3547*  3587*  3635*  3679*  3692*  3703*  3749*  3770*
                    3771*  3786*  3787*  3827*  3839*  3855*  3856*  3861*  3865*  3870   3888*  3889*  3896*
                    3899*  3911   3932*  3933*  3937*  3939*  3943*  3967   3991*  4042*  4082*  4099*  4100*
                    4101*  4102*  4103*  4146*  4163*  4164*  4165*  4166*  4167*  4188*  4195   4232*  4233
                    4264*  4265   4280*  4282   4299*  4305*  4306*  4307   4311   4319*  4320*  4321   4325
                    4333*  4334*  4335   4339   4382*  4395*  4396*  4424*  4425*  4426*  4457*  4458*  4459*
                    4460   4461*  4462   4467   4469*  4480*  4481*  4483   4552*  4571*  4572*  4642*  4643*
                    4647   4652   4653*  4667*  4677*  4679   4694*  4696   4700*  4722*  4723*  4724*  4725
                    4726*  4727*  4728   4732   4733*  4742   4746   4786*  4787*  4788   4818*  4819*  4820
                    4835*  4836*  4837   4856*  4857*  4891*  4892*  4905*  4941*  4961*  4962*  4989*  4993*
                    4995*  5006*  5007   5008*  5009   5020   5032   5044   5055   5097   5101*  5102*  5103*
                    5104*  5105*  5106*  5107   5111*  5143*  5149*  5153*  5238*  5239*  5241   5257*  5258*
                    5260   5272*  5285*  5324*  5331*  5335*
R3      =X000003    2258#  2681#  2714#  2715   2724#  2746#  2772#  2773   2784#  2785   2796#  2797   2821#
                    2822   2831#  2855#  2856   2865#  2889#  2890#  2891   2900#  2901#  2902   2928#  2929
                    2939#  2940   2966#  2967   2977#  2978   3004#  3005#  3006   3016#  3017#  3018   3052#
                    3053   3098#  3099   3132#  3133   3143#  3166#  3167   3177#  3212#  3216   3258#  3262
                    3303#  3307   3348#  3352   3398#  3399#  3400#  3404   3424#  3425#  3426#  3427   3428#
                    3429#  3447#  3454#  3468#  3469#  3470#  3471   3472#  3473#  3484#  3493#  3508#  3509#
                    3510#  3511   3512#  3513#  3524#  3533#  3548#  3549#  3550#  3551   3552#  3553#  3564#
                    3573#  3588#  3589#  3590#  3591   3592#  3593#  3604#  3613#  3627#  3628#  3629#  3630#
                    3656#  3657#  3669#  3675#  3686#  3687   3698#  3710#  3711#  3712#  3713#  3714#  3746#
                    3760#  3764   3776#  3780   3800#  3801#  3802#  3809#  3810#  3814#  3821#  3823   3832#
                    3833#  3845   3846#  3847#  3848#  3849#  3862#  3863#  3870#  3871#  3873   3894#  3900#
                    3903#  3908#  3911#  3912#  3915   3940#  3941#  3967#  4008#  4009#  4028#  4029#  4043#
                    4080#  4081#  4092#  4094   4145#  4156#  4158   4207#  4208#  4218#  4223   4265#  4266#
                    4288#  4289   4311#  4325#  4339#  4379#  4390#  4391   4456#  4457   4462   4553#  4554#
                    4565#  4566   4626#  4639   4647   4663   4695#  4699#  4721#  4722   4728   4788#  4789#
                    4820#  4837#  4838#  4850#  4851   4890#  4906#  4907#  4937#  4938#  4954#  4957   4988#
                    4989#  4992#  4995#  5033   5056   5144#  5150   5152#  5298#  5299#  5300#  5301#  5302#
                    5303#  5304#  5325#  5332   5334#
R4      =X000004    2258#  2682#  2710#  2715   2721#  2742#  2766#  2767#  2773   2781#  2785   2792#  2793
                    2797   2817#  2822   2828#  2851#  2856   2862#  2885#  2891   2897#  2902   2923#  2925#
                    2929   2936#  2940   2960#  2961#  2967   2974#  2978   2990#  2999#  3006   3013#  3018
                    3036#  3040   3041#  3048#  3053   3063#  3082#  3086   3087#  3094#  3099   3109#  3124#
                    3131   3133   3140#  3142   3158#  3165   3167   3174#  3176   3209#  3215#  3216   3255#
                    3261#  3262   3300#  3306#  3307   3345#  3351#  3352   3394#  3403#  3404   3440#  3487#
                    3494#  3527#  3534#  3567#  3574#  3607#  3614#  3636#  3637#  3680#  3691#  3702#  3747#
                    3748#  3759#  3763#  3764   3775#  3779#  3780   3815#  3822#  3823   3834#  3835   3860#
                    3866#  3872#  3873   3895#  3897#  3904#  3906#  3913#  3914#  3915   3968#  3969#  3970#
                    3971#  3972#  4006#  4007#  4026#  4027#  4041#  4083#  4084#  4093#  4094   4147#  4148#
                    4157#  4158   4209#  4210#  4219#  4220#  4221#  4222#  4223   4267#  4268#  4282#  4283#
                    4284#  4285#  4286#  4287#  4289   4312#  4326#  4340#  4359#  4360#  4361#  4380#  4381#
                    4391   4467#  4527#  4555#  4556#  4566   4639#  4640#  4641#  4642   4644   4651   4652#
                    4668#  4732#  4760#  4785   4790#  4791#  4796#  4817   4821#  4826#  4834   4839#  4840#
                    4845#  4851   4893#  4908#  4939#  4940#  4950#  4951#  4952#  4953#  4957   4988#  4995#
                    5034   5057   5145#  5150#  5154#  5157#  5158#  5159   5326#  5332#  5336#
R5      =X000005    2258#  2637#  2650#  2651#  2658   2673#  2674#  2675#  2676#  2677#  2678#  2679#  2680#
```

```
                    2681    2684    2685    2702*   2704    2706*   2710    2713    2723    2735*   2736    2738*   2745
                    2759*   2760    2762*   2766    2771    2783    2795    2810*   2811    2813*   2817    2820    2830
                    2844*   2845    2847*   2851    2854    2864    2878*   2879    2881*   2885    2888    2899    2915*
                    2916    2918*   2925    2927    2936    2938    2953*   2954    2956*   2961    2965    2974    2976
                    2991*   2992    2994*   2999    3003    3013    3015    3057*   3103*   3126*   3160*   3190*   3191
                    3220*   3236*   3237    3266*   3282*   3283    3290    3306    3311*   3327*   3328    3335    3351
                    3356*   3372*   3373    3374    3408*   3430*   3431    3433*   3440    3441    3659*   3719*   3751*
                    3820*   3832    3859*   3867*   3892*   3893*   3913    3936*   3937    3938*   3944*   3947    3956
                    3960    4011*   4019*   4020*   4031*   4045*   4086*   4105*   4150*   4169*   4187*   4189    4212*
                    4227*   4251*   4270*   4276*   4277    4293*   4314*   4328*   4342*   4384*   4398*   4408*   4410*
                    4412    4416    4417    4471*   4474*   4520*   4527    4535    4558*   4574*   4589*   4597*   4604
                    4615    4633*   4656*   4671*   4693*   4702*   4736*   4753*   4760    4793*   4823*   4842*   4859*
                    4895*   4910*   4943*   4964*   4988*   4989*   4995*   5098    5099    5112*
R6      =X000006    2258#   2596*   5021    5035
R7      =X000007    2258#
SAPS    024334      3446    3453    3483    3492    3523    3532    3563    3572    3603    3612    3655    3668    3678
                    3690    3701    3745    3813    3826    3838    3950    3959    4005    4025    4040    4098    4162
                    4206    4263    4310    4324    4338    4378    4394    4465    4550    4569    4650    4666    4731
                    4784    4816    4833    4855    4889    4904    4936    4960    5016    5028    5363#
SELINE  023024      3032    3044    3078    3090    3122    3156    3193    3239    3284    3329    3386    3631    3728
                    3988    4056    4121    4185    4241    4356    4530    4607    4763    4871    4920    5070#
SELMSK  024404      2623*   2629    2631*   4978*   4980    5389#
SETALL  024174      4612    5323#
SP      =X000006    2258#   2596*   2652*   2663    2668    3437*   3450*   3457*   3460*   3475*   3500*   3515*   3540*
                    3555*   3580*   3595*   3620*   3634*   3716*   3804*   3843*   4414*   4435*   4984*   4985*   4986*
                    4987*   4988*   4989*   4990*   4991*   4992*   4993*   4994*   4995*   5081*   5097*   5111    5170
                    5193    5201    5209    5216    5237*   5[illegible]*   5270    5271    5272    5274*   5283    5284    5285
                    5287*   5317*   5347*   5348*   5355*   5[illegible]*   5[illegible]*   5364*   5367*   5369*   5370    5371
SSR     = 000016    2678    2712*   2722*   2744*   2770*   2[illegible]*   2819*   2829*   2853*   2863*   2877    2924*   2937*
                    2964*   2975*   2990    3039*   3051*   30[illegible]*   3097*   3130*   3141*   3164*   3175*   3205*   3211*
                    3251*   3257*   3296*   3302*   3341*   33[illegible]*   3371    3[illegible]6*   3[illegible]0    3839    3856    3889    3933
                    4037    4042    4043    4683    4778    4787    4830    4836    5[illegible]5#
STACK = 001100      2258#   2596    3437    3460    3475    3500    3515    3540    3555    3580    3595    3620    3634
                    3716    3804    3843    4414    4435    5274    5287
START1  002400      2599#
START2  002506      2613    2619#   2632    4984    5226
STKLMT= 177774      2258#
SUER1   022632      2662    5016#
SUER2   022672      2692    2718    2727    2749    2776    2788    2800    2825    2834    2859    2868    2894    2905
                    2932    2943    2970    2981    3009    3021    3056    3102    3136    3146    3170    3180    3407
                    3876    3918    4226    4292    5028#
SUER2A  022676      3448    3455    3463    3497    3528    3537    3568    3577    3608    3617    3658    3670    3681
                    3693    3704    3750    3816    3828    3840    3952    3963    4010    4030    4[illegible]2    4085    4104
                    4149    4168    4211    4269    4313    4327    4341    4383    4397    4470    4557    4573    4655
                    4670    4735    4792    4[illegible]2    4841    4858    4894    4909    4942    4963    5029#
SUER3   022742      5042#   5275    5288
SUER4   022760      3219    3265    3310    3355    3772    3788    5050#
SUNUM   023116      2637    3057    3103    3126    3160    3220    3266    3311    3356    3408    3659    3719    3751
                    4011    4031    4045    4086    4105    4150    4169    4212    4227    4270    4293    4314    4328
                    4342    4384    4398    4471    4474    4558    4574    4633    4656    4671    4736    4793    4823
                    4842    4859    4895    4910    4943    4964    5097#
SUPPAR  023300      4420    5141#
SWR     001136      2259#   2596*   4985    4386    5221
SWREG   000176      2247#   2596
```

```
SW0     = 000001          2258#
SW00    = 000001          2258#
SW01    = 000002          2258#
SW02    = 000004          2258#
SW03    = 000010          2258#
SW04    = 000020          2258#
SW05    = 000040          2258#
SW06    = 000100          2258#
SW07    = 000200          2258#
SW08    = 000400          2258#
SW09    = 001000          2258#
SW1     = 000002          2258#
SW10    = 002000          2258#
SW11    = 004000          2258#
SW12    = 010000          2258#
SW13    = 020000          2258#
SW14    = 040000          2258#
SW15    = 100000          2258#
SW2     = 000004          2258#
SW3     = 000010          2258#
SW4     = 000020          2258#
SW5     = 000040          2258#
SW6     = 000100          2258#
SW7     = 000200          2258#
SW8     = 000400          2258#
SW9     = 001000          2258#
TBITVE= 000014            2258#
TBUF    033436            4248    4280    4335    4340    4802    4876    4924    5005    5146    5328    5852#
TCRLF = 000200            4990#
TDATA1  025156            4363#   4366    5603#
TDATA2  025160            4187    5604#
THT   = 000011            4990#
TIMEA   025172            3648#   3737#   3998#   4067#   4132#   4198#   4255#   4370#   4488#   4502#   4542#   4624#   4709#
                          4776#   4878#   4891#   4929#   5315#   5609#
TIMEB   025174            3649#   3738#   3999#   4068#   4092    4110    4133#   4156    4174    4199#   4256#   4371#   4489#
                          4503#   4543#   4625#   4710#   4777#   4809#   4882#   4930#   5313#   5610#
TIMEC   025176            4059#   4093    4110#   4124#   4157    4174#   5611#
TIMEIT  024152            3651    3741    4002    4071    4136    4202    4259    4374    4492    4506    4546    4628    4713
                          4780    4812    4845    4933    5313#
TITFLG  025170            2594#   2603    2607#   5608#
TITLE   031566            2606    5828#
TITLE2  031636            2639    2641    5829#
TKVEC = 000060            2258#
TNULL   025200            4771    5612#
TPVEC = 000064            2258#
TRAPVE= 000034            2258#   2596#
TRTVEC= 000014            2258#
TST1    002662            2647#
TST10   003734            2848    2873#
TST11   004070            2882    2910#
TST12   004226            2919    2948#
TST13   004364            2957    2986#
TST14   004532            2995    3026#
TST15   004730            3072#
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
TST16   005126          3118#
TST17   005260          3123    3152#
TST2    002760          2671#
TST20   005412          3157    3186#
TST21   005562          3192    3232#
TST22   006060          3238    3278#
TST23   006250          3285    3323#
TST24   006450          3330    3368#
TST25   006672          3375    3420#
TST26   007116          3464#
TST27   007340          3504#
TST3    003036          2686    2698#
TST30   007746          3544#
TST31   010076          3584#
TST32   010306          3624#
TST33   010506          3717    3725#
TST34   011102          3729    3797#
TST35   011362          3853#
TST36   011430          3886#
TST37   011614          3929#
TST4    003204          2707    2732#
TST40   012036          3985#
TST41   012354          3989    4053#
TST42   012670          4057    4118#
TST43   013200          4122    4182#
TST44   013446          4186    4238#
TST45   014126          4353#
TST46   014366          4357    4405#
TST47   015100          4438    4515#
TST5    003274          2739    2755#
TST50   015460          4587    4594#
TST51   016126          4608    4689#
TST52   016434          4719    4740    4750#
TST53   017202          4868#
TST54   017424          4872    4917#
TST6    003470          2763    2805#
TST7    003612          2814    2839#
TYPDS = 104404          4984    4994#
TYPE  = 104400          2605    2640    4984    4986    4987    4988    4989    4990    4992    4993    4994#   4995    5167
                        5190    5198    5206    5213    5223    5244    5263
TYPOC = 104401          4987    4994#
TYPON = 104403          4994#
TYPOS = 104402          4994#
VCFLG   025076          2608    2611#   5187#   5569#
VCPTR   025152          2620#   2627#   2635    4976#   5600#
VCWC    032306          5168    5838#
$APTHD  000214          2252#
$ASTAT= ****** U        4991
$ATYC   021646          4991#
$ATY1   021622          4991#
$ATY3   021630          4990    4991#
$ATY4   021640          4986    4991#
$BASE   001304          2259#
$BOROR  001122          2259#
```

DZDMMA.P11 CROSS REFERENCE TABLE

```
$BDOAT  001126          2259#   259%
$CDM1   001310          2259#
$CDM2   001312          2259#
$CHARC  021616          4990#*
$CMTAG  001100          2259#   259%
$CM1  = 000010          2259#
$CM2  = 000020          2259#
$CM3  = 000010          2259#
$CM4  = 000010          2259#
$CPUOP  001256          2259#
$CRLF   001225          2259#   4986    4987    4990    4992    4993
$DBLK   021334          4989#
$DDM0   001314          2259#
$DDM1   001316          2259#
$DDM10  001340          2259#
$DDM11  001342          2259#
$DDM12  001344          2259#
$DDM13  001346          2259#
$DDM14  001350          2259#
$DDM15  001352          2259#
$DDM2   001320          2259#
$DDM3   001322          2259#
$DDM4   001324          2259#
$DDM5   001326          2259#
$DDM6   001330          2259#
$DDM7   001332          2259#
$DDM8   001334          2259#
$DDM9   001336          2259#
$DEVCT  001240          2259#
$DEVM   001306          2259#
$DOAGN  020056          4984#
$DTBL   021324          4989#
$ENDAD  020046          2251    4984#
$ENDCT  020014          4984#
$ENDMG  020062          4984#
$ENULL  020077          4984#
$ENV    001250          2259#   4986    4990    4991
$ENVM   001251          2259#   259%    4990    4991
$EOP    017760          4973#   4979    4984#
$EOPCT  020006          4984#
$ERFLG  001103          2259#   4985#   4986#
$ERMAX  001115          2259#   259%#   4985#
$ERROR  020370          259%    4986#
$ESAPC  001116          2259#   4986#   4987    5626    5633    5722    5738    5753    5792
$ESRTB  001354          2259#   4987
$ESRTY  020536          4986    4987#
$ERTTL  001112          2259#   4986#
$ESCAP  001222          2259#   259%#   4985#   4986
$ETABL  001250          2259#
$ETEND  001354          2252    2259#
$FATAL  001232          2259#   4991#
$FFLG   022066          4991#*
$FILLC  001154          2259#   4990
$FILLS  001153          2259#   4990
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
$GDADR  001120          2259#
$GDAT   001124          2259#
$GET42  020036          4984#
$HO   = 000001          2244
$HIBTS  000214          2252#
$HIOCT  022400          4993#*
$ICNT   001104          2259#   4985*
$ILLUP  022572          4995#
$ITEMB  001114          2259#   4986*   4987
$LF     001226          2259#   4986    4990    4992    4993
$LFLG   022065          4991#*
$LPADR  001106          2259#   2596*   2644*   4985*
$LPERR  001110          2259#   2596*   2664*   2693*   2700*   2750*   2757*   2807*   2841*   2875*   2912*   2950*   2988*
                        3060*   3106*   3137*   3147*   3171*   3181*   3223*   3269*   3314*   3359*   3411*   3422*   3466*
                        3506*   3546*   3586*   3626*   3727*   3799*   3877*   3919*   3931*   3987*   4055*   4120*   4184*
                        4240*   4355*   4407*   4517*   4596*   4691*   4752*   4870*   4919*   4985*   4986    5276*   5289*
$MADR1  001262          2259#
$MADR2  001266          2259#
$MADR3  001272          2259#
$MADR4  001276          2259#
$MAIL   001230          2252    2259#   2596    2647    2671    2698    2732    2755    2805    2839    2873    2910    2948
                        2986    3026    3072    3118    3152    3186    3232    3278    3323    3368    3420    3464    3504
                        3544    3584    3624    3725    3797    3853    3886    3929    3985    4053    4118    4182    4238
                        4353    4405    4515    4594    4689    4750    4868    4917    4985    4986    4990
$MAMS1  001260          2259#
$MAMS2  001264          2259#
$MAMS3  001270          2259#
$MAMS4  001274          2259#
$MBADR  000216          2252#
$MFLG   022064          4991#*
$MSGAD  001244          2259#   4991*
$MSGLG  001246          2259#   4991*
$MSGTY  001230          2259#   4991*
$MTYP1  001261          2259#
$MTYP2  001265          2259#
$MTYP3  001271          2259#
$MTYP4  001275          2259#
$MXCNT  020366          4985#
$NULL   001152          2259#   4990
$NWTST= 000001          2647#   2671#   2698#   2732#   2755#   2805#   2839#   2873#   2910#   2948#   2986#   3026#   3072#
                        3118#   3152#   3186#   3232#   3278#   3323#   3368#   3420#   3464#   3504#   3544#   3584#   3624#
                        3725#   3797#   3853#   3886#   3929#   3985#   4053#   4118#   4182#   4238#   4353#   4405#   4515#
                        4594#   4689#   4750#   4868#   4917#
$OCNT   021114          4988#*
$OMODE  021116          4988#*
$OVER   020352          4985#
$PASS   001236          2259#   2596*   4984*   4985
$PASTM  000222          2252#
$POWER  022600          4995#
$PRIOR  001302          2259#
$PWRAD  022566          4995#
$PWRON  022444          2596    4995#
$PWRMG  022562          4995#
$PWRUP  022512          4995#
```

```
$QUES   001224          2259#  4986   4990   4992   4993
$RDCHR  022070          4992#  4994
$RDDEC= ****** U        4994
$RDLIN  022124          4992#  4994
$RDOCT  022242          4993#  4994
$RDSZ = 000010          4992#
$REGAD  001156          2259#
$REG0   001160          2259#  5018*  5030*  5042*  5053*  5626   5633   5722   5792
$REG1   001162          2259#  5019*  5031*  5043*  5054*  5626   5633   5722   5792
$REG2   001164          2259#  5020*  5032*  5044*  5055*  5626   5633   5722   5792
$REG3   001166          2259#  5033*  5056*  5633   5792
$REG4   001170          2259#  5034*  5057*  5633   5792
$REG5   001172          2259#
$REG6   001174          2259#  5021*  5022*  5035   5036*  5270*  5283*  5626   5633   5722
$REG7   001176          2259#
$R2A  = ****** U        4994
$SAVRE= ****** U        4994
$SAVR6  022576          4995#*
$SCOPE  020102          2596   4985#
$SETUP= 000017          2595#  2596   4984   4986
$STUP = 177777          2595#
$SVLAD  020316          4985#
$SVPC = 000214          2251#
$SWR  = 165400          2237#  2244   2245   2259   2596   2647   2671   2698   2732   2755   2805   2839   2873
                        2910   2948   2986   3026   3072   3118   3152   3186   3232   3278   3323   3368   3420
                        3464   3504   3544   3584   3624   3725   3797   3853   3886   3929   3985   4053   4118
                        4182   4238   4353   4405   4515   4594   4689   4750   4868   4917   4984   4985   4986
                        4995
$SWREG  001252          2259#  2596
$SWRMK= 000000          2245   4985
$TESTN  001234          2259#  2647*  2671*  2698*  2732*  2755*  2805*  2839*  2873*  2910*  2948*  2986*  3026*
                        3072*  3118*  3152*  3186*  3232*  3278*  3323*  3368*  3420*  3464*  3504*  3544*  3584*
                        3624*  3725*  3797*  3853*  3886*  3929*  3985*  4053*  4118*  4182*  4238*  4353*  4405*
                        4515*  4594*  4689*  4750*  4868*  4917*  4985*
$TIMES  001220          2259#  2596*  4984*  4985*
$TKB    001144          2259#  4992
$TKS    001142          2259#  4992
$TMP0   001200          2259#  3767*  3768*  3783*  3784*  4075*  4076*  4077*  4078*  4079*  4140*  4141*  4142*
                        4143*  4144*  4421*  4632*  4717*  5050*  5051*  5269*  5282*  5371*  5626   5633   5722
                        5738   5753   5792
$TMP1   001202          2259#  3191*  3206   3209   3237*  3252   3255   3283*  3328*  3373*  3389   3769*  3785*
                        4785*  4796   4817*  4826   4834*  4845   4949*  4955*  5792
$TMP2   001204          2259#  3208*  3210   3213   3221   3226*  3227   3254*  3256   3259   3267   3272*  3273
                        3299*  3301   3304   3312   3317*  3318   3344*  3346   3349   3357   3362*  3363   3374*
                        3394   4430*  4431*  4619*  4620*  4704*  4705*  5738
$TMP3   001206          2259#  3197*  3198   3200*  3201   3243*  3244   3246*  3247   3288*  3289   3291*  3292
                        3333*  3334   3336*  3337   3377*  3379   3382*  3383   4422*  4446*  4510*  4518*  4586*
                        5753
$TMP4   001210          2259#  3393*  3396   3401   3409   3414*  3415   4460*  4472   5141*  5155*
$TMP5   001212          2259#  4417*  4651*  4657   5148
$TMP6   001214          2259#  4416*  4538*  4554*  4725*  4737   4744   5147   5158   5323*  5337*
$TMP7   001216          2259#  3030*  3069   3076*  3115   3757*  3758   3761   3769   3777   3795   3791*  3792
                        3857*  3859   3880*  3881   3890*  3893   3903   3922*  3923   3934*  3936   3968   3972
                        3975   3980*  3981   4017*  4016*  4058*  4064   4075   4099   4111*  4112   4123*  4129
```

```
                        4140    4163    4175*   4176    4189*   4193    4222    4409*   4411*   4418*   4475    4537*   4580*
                        4604*   4663    4667    4668    5117*   5118    5121*   5122    5130*   5131    5133*   5134
$TN    = 000055         2244*   2647*   2671*   2686    2698*   2707    2732*   2739    2755*   2763    2805*   2814    2839*
                        2848    2873*   2882    2910*   2919    2948*   2957    2986*   2995    3026*   3072*   3118*   3123
                        3152*   3157    3186*   3192    3232*   3238    3278*   3285    3323*   3330    3368*   3375    3420*
                        3464*   3504*   3544*   3584*   3624*   3717    3725*   3729    3797*   3853*   3886*   3929*   3985*
                        3989    4053*   4057    4118*   4122    4182*   4186    4238*   4353*   4357    4405*   4438    4515*
                        4587    4594*   4608    4689*   4719    4740    4750*   4868*   4872    4917*
$TPB    001150          2259*   4990*
$TPFLG  001155          2259*   4990
$TPS    001146          2259*   4990
$TRAP   022402          2596    4994*
$TRP   = 000010         4994*
$TRPAD  022424          4994*
$TSTM   000220          2252*
$TSTNM  001102          2259*   4430    4619    4704    4982*   4984*   4985*   4986    5017    5029    5052    5273    5286
$TTYIN  022232          4992*
$TYPBN= ****** U        4994
$TYPOS  021120          4989*   4994
$TYPE   021344          4990*   4991    4994
$TYPEC  021552          4990*
$TYPEX  021620          4990*
$TYPOC  020716          4988*   4994
$TYPON  020732          4988*   4994
$TYPOS  020672          4988*   4994
$UNIT   001242          2259*
$UNITM  000224          2252*
$USWR   001254          2259*
$VECT1  001300          2259*
$VECT2  001301          2259*
$XTSTR  020122          4985*
$DFILL  021115          4988**
$40CAT= ****** U        4985    4986
.       = 034036        2247*   2251*   2252*   2259*   2596    2644    4984    4985    4986    4987*   4989*   4990    4991*
                        4992*   4993    4995    5395*   5499*   5632*   5752*   5842*   5846*   5852*
.$ASTA= ****** U        4991
.$X    = 000214         2252*
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
COMMEN  2258#
ENDCOM  2258#
ERROR   2258#  2665  2694  2719  2728  2751  2777  2789  2801  2826  2835  2860  2869  2895  2906
        2933   2944  2971  2982  3010  3022  3061  3107  3138  3148  3172  3182  3224  3270  3315
        3360   3412  3449  3456  3489  3498  3529  3538  3569  3578  3609  3618  3662  3672  3683
        3695   3706  3754  3773  3789  3817  3829  3841  3878  3920  3953  3964  4014  4034  4048
        4089   4108  4153  4172  4215  4230  4273  4296  4317  4331  4345  4387  4401  4446  4453
        4477   4498  4512  4561  4577  4636  4659  4674  4718  4739  4797  4827  4846  4862  4898
        4913   4946  4967  5277  5290
ESCAPE  2258#
MHORS0  1622#  3986
MHOR1      3#  2648
MHOR10   350#  2874
MHOR11   405#  2911
MHOR12   448#  2949
MHOR13   491#  2987
MHOR14   534#  3027
MHOR15   633#  3073
MHOR16   732#  3119
MHOR17   773#  3153
MHO[illegible]2     45#  2672
MHO[illegible]20   814#  3187
MHO[illegible]21   859#  3233
MH[illegible]22   904#  3279
[illegible]3   9[illegible]#  3324
[illegible]4   9 [illegible]#  3369
[illegible]5  10[illegible]#  3421
[illegible]   11[illegible]#  3465
[illegible]7  11[illegible]#  3505
[illegible]3    9 #  2699
[illegible]0  11[illegible]#  3545
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
.SERRT   2240#   4987
.SPOWE   2240#   4995
.SRDOC   2241#   4993
.SREAD   2241#   4992
.SSCOP   2240#   4985
.STRAP   2240#   4994
.STYPD   2240#   4989
.STYPE   2240#   4990
.STYPO   2240#   4988
```

```
ADD     2626  2627  2628  2651  2656  2678  2679  2680  2809  2843  2877  2914  2952  2990  3029
        3041  3063  3075  3087  3109  3121  3155  3189  3235  3281  3326  3371  3628  3679  3692
        3703  3711  3713  3771  3787  3827  3839  3848  3856  3889  3933  4042  4111  4175  4300
        4306  4320  4334  4396  4468  4469  4572  4653  4700  4733  4734  4787  4819  4836  4857
        4893  4962  4975  4976  4977  4987  4988  4989  4990  4991  4993  5022  5036  5081  5106
        5109  5153  5154  5237  5240  5256  5259  5300  5303  5317  5335  5336
ASL     2631  2706  2738  2762  2813  2847  2881  2918  2956  2994  3433  4141  4142  4164  4165
        4461  4643  4726  4978  4987  4993  4994  5077
ASLB    4989
ASR     4077  4078  4101  4102  4991  5102  5103  5104
BCC     4989
BEQ     2596  2609  2613  2632  2686  2690  2707  2716  2725  2739  2747  2763  2774  2786  2798
        2814  2823  2832  2848  2857  2866  2882  2892  2903  2919  2930  2941  2957  2968  2979
        2995  3007  3019  3054  3100  3134  3144  3168  3178  3192  3214  3217  3228  3238  3260
        3263  3274  3305  3308  3319  3350  3353  3364  3375  3402  3405  3416  3434  3676  3688
        3699  3765  3781  3824  3836  3874  3916  3957  3961  3982  4023  4038  4224  4290  4308
        4322  4336  4392  4413  4450  4463  4491  4505  4526  4529  4567  4587  4603  4606  4645
        4648  4664  4729  4759  4762  4779  4811  4831  4884  4958  4979  4981  4984  4985  4986
        4987  4998  4990  4991  4993  5078  5080  5171  5173  5175  5194  5202  5210  5217
BGE     4985  4990  5230  5249
BGT     4581  4984  4988  4989  4993
BHI     4985
BIC     2723  2780  2781  2783  2795  2830  2864  2890  2899  2901  2925  2927  2961  2965  2999
        3003  3005  3017  3400  3657  3833  3865  3871  3899  3912  3933  4009  4029  4079  4081
        4103  4144  4167  4208  4266  4283  4431  4459  4483  4554  4583  4620  4621  4641  4679
        4682  4705  4724  4745  4746  4789  4838  4907  4938  4984  4988  4992  4993  5105  5108
BIC8    3378  3380  3395  5084  5124  5136
BIS     2688  2767  2769  2771  2793  2854  2888  2936  2938  2974  2976  3013  3015  3441  3478
        3479  3518  3519  3558  3559  3598  3599  3644  3645  3730  3806  3861  3896  3939  4195
        4252  4940  4953  4988  4989
BISB    3038  3050  3084  3096  3125  3159  3379  3388  3389  3396  3637  3641  3733  3748  3972
        4007  4027  4061  4084  4126  4148  4191  4196  4210  4220  4222  4247  4268  4285  4286
        4360  4362  4381  4556  4614  4875  4952  4987
BIT     2629  2704  2736  2760  2811  2845  2879  2916  2954  2992  3431  3981  4000  4022  4449
        4528  4605  4761  4980  4985  4986  5079  5235  5254
BITB    2596  4940  4991
BLO     4095  4159
BLOS    4992
BLT     4988  4989  4990  4993  5233  5252
BMI     3646  3740  3946  4070  4135  4201  4258  4373  4443  4545  4582  4627  4712  4932  4989
BNE     2596  2604  2615  2630  2659  2705  2737  2761  2812  2846  2880  2917  2955  2993  3035
        3047  3081  3093  3202  3248  3293  3338  3384  3432  3762  3778  3793  3811  3864  3868
        3882  3898  3901  3907  3909  3924  3942  3948  3976  4001  4021  4113  4177  4234  4279
        4302  4482  4678  4681  4684  4703  4743  4747  4852  4956  4985  4986  4987  4988  4989
        4990  4991  4992  4995  5010  5071  5123  5135  5156  5160  5236  5242  5255  5261  5314
        5316  5338
BPL     4902  4986  4988  4989  4990  4992  5222
BR      2596  2633  2660  2667  2695  2708  2729  2740  2752  2764  2778  2790  2802  2815  2836
        2849  2870  2883  2907  2920  2934  2945  2958  2972  2983  2996  3011  3023  3033  3042
        3045  3064  3067  3079  3088  3091  3110  3113  3123  3149  3157  3163  3194  3229  3240
        3275  3285  3320  3330  3365  3387  3417  3435  3444  3490  3530  3570  3610  3632  3652
        3663  3673  3684  3696  3707  3717  3729  3742  3755  3794  3818  3830  3954  3965  3978
        3989  4003  4015  4035  4050  4057  4072  4090  4114  4122  4137  4154  4178  4186  4203
        4216  4235  4242  4243  4260  4274  4357  4375  4388  4402  4438  4447  4454  4478  4493
```

```
        4507  4523  4531  4547  4562  4578  4584  4600  4608  4629  4637  4714  4719  4740  4756
        4764  4765  4781  4798  4813  4828  4872  4886  4899  4914  4921  4934  4947  4968  4985
        4986  4987  4988  4989  4990  4991  4992  4993  4995  5075  5082  5176  5178  5196  5204
        5219  5231  5234  5243  5250  5253  5262
CLC     3979  4524  4601  4757
CLR     2593  2594  2596  2611  2682  2711  2721  2742  2743  2768  2818  2828  2852  2862  2886
        2897  2923  2962  3000  3036  3048  3066  3092  3094  3112  3140  3174  3199  3215  3245
        3261  3297  3300  3342  3345  3381  3403  3485  3486  3495  3496  3525  3526  3535  3536
        3565  3566  3575  3576  3605  3606  3615  3616  3649  3680  3702  3714  3738  3757  3763
        3767  3775  3783  3845  3849  3860  3934  3993  3999  4026  4063  4068  4128  4133  4188
        4199  4256  4312  4326  4371  4409  4489  4503  4518  4521  4538  4543  4551  4570  4588
        4598  4625  4699  4710  4754  4777  4809  4821  4882  4893  4930  4984  4985  4987  4988
        4989  4993  4995  5006  5050  5083  5117  5119  5120  5130  5143  5301  5304  5324  5327
        5340  5363
CLRB    3197  3208  3243  3254  3288  3299  3333  3344  3377  3393  3971  4982  4985  4989  4990
        4991  4992  4993  5074  5142  5149  5161  5331
CMP     2596  2614  2658  2663  2685  2715  2773  2785  2797  2822  2856  2891  2902  2929  2940
        2967  2978  3006  3018  3053  3099  3133  3167  3216  3262  3307  3352  3404  3687  3764
        3780  3792  3823  3835  3873  3881  3915  3923  3960  4094  4112  4158  4176  4223  4233
        4277  4289  4301  4335  4391  4412  4462  4566  4647  4663  4728  4742  4851  4957  4985
        4989  4992  5009  5122  5134  5159  5172  5174  5229  5232  5241  5248  5251  5260
CMPB    3201  3213  3227  3247  3259  3273  3292  3304  3318  3337  3349  3363  3383  3401  3415
        3761  3777  4037  4644  4778  4830  4985  4986  4990  4991  4992  4993
COM     2607  4586
COMB    5072
DEC     3810  3863  3867  3897  3900  3906  3908  3941  3944  4020  4411  4580  4702  4955  4984
        4987  5155  5315  5337
DECB    4988  4990
EMT     2258
HALT    2247  4986  4990  4995  5225
INC     2625  3779  3791  3866  3880  3922  3938  3969  4232  4299  4418  4481  4564  4701  4727
        4974  4984  4985  4986  4988  4989  4991  4995  5008  5121  5133  5151  5152  5313  5333
        5334
INCB    3200  3226  3246  3272  3291  3317  3336  3362  3382  3414  4480  4677  4985  4986  4990
        5076
IOT     2258
JMP     2247  2248  2249  2616  2617  4244  4347  4433  4499  4513  4590  4660  4675  4685  4686
        4766  4847  4853  4863  4983  4984  5185  5198  5226  5290  5293
JSR     2610  2637  2642  2662  2692  2718  2727  2779  2776  2798  2800  2825  2834  2859  2868
        2894  2905  2932  2943  2970  2981  3009  3021  3032  3044  3056  3057  3078  3090  3102
        3103  3122  3126  3136  3146  3156  3160  3170  3190  3193  3219  3220  3239  3265  3266
        3284  3310  3311  3329  3335  3356  3396  3407  3428  3438  3442  3446  3448  3453  3455
        3461  3476  3480  3483  3488  3492  3497  3501  3516  3520  3523  3528  3532  3537  3541
        3556  3560  3563  3568  3572  3577  3581  3596  3600  3603  3609  3612  3617  3621  3631
        3638  3639  3646  3651  3655  3658  3659  3668  3670  3671  3678  3681  3682  3690  3693
        3694  3701  3704  3705  3715  3719  3728  3731  3732  3741  3745  3750  3751  3772  3788
        3003  3808  3813  3816  3826  3828  3838  3840  3844  3876  3918  3950  3951  3952  3959
        3962  3963  3968  4002  4005  4010  4011  4025  4030  4031  4040  4044  4045  4056  4071
        4085  4086  4098  4104  4105  4121  4136  4149  4150  4162  4168  4169  4185  4202  4206
        4211  4212  4226  4227  4241  4259  4263  4269  4270  4292  4293  4310  4313  4314  4324
        4327  4328  4338  4341  4342  4356  4374  4378  4383  4384  4394  4397  4398  4420  4423
        4432  4436  4437  4465  4470  4471  4474  4492  4506  4530  4532  4546  4550  4557  4558
        4569  4573  4574  4607  4612  4628  4633  4650  4655  4656  4666  4670  4671  4713  4731
        4735  4736  4763  4768  4780  4784  4792  4793  4812  4816  4822  4823  4833  4841  4842
```

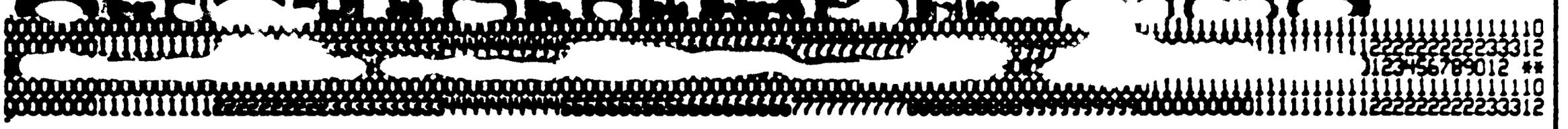
```
                4855    4858    4859    4871    4874    4885    4889    4894    4895    4904    4909    4910    4920    4933    4936
                4942    4943    4960    4963    4964    4984    4986    4990    4991    5016    5028    5195    5203    5275    5279
                5288    5292
MOV             2596    2598    2600    2601    2602    2619    2620    2621    2622    2623    2634    2635    2636    2644    2647
                2649    2650    2652    2653    2664    2668    2671    2673    2674    2675    2676    2677    2684    2687    2689
                2693    2698    2700    2701    2702    2710    2713    2714    2724    2733    2734    2735    2745    2746    2750
                2755    2757    2758    2759    2766    2772    2784    2792    2794    2796    2805    2807    2808    2810    2917
                2820    2821    2831    2839    2841    2842    2844    2851    2855    2865    2873    2875    2876    2878    2885
                2887    2889    2898    2900    2910    2912    2913    2915    2922    2928    2933    2948    2950    2951
                2953    2960    2963    2966    2977    2986    2988    2989    2991    3001    3002    3004    3014    3016
                3026    3028    3030    3031    3040    3058    3060    3069    3072    3074    3077    3086    3098    3106
                3115    3118    3120    3124    3131    3132    3137    3142    3143    3147    3152    3158    3165    3166
                3171    3176    3177    3181    3186    3188    3190    3191    3201    3210    3211    3232    3234    3236
                3237    3252    3255    3258    3269    3278    3290    3293    3303    3313    3314    3322    3323    3325
                3327    3328    3335    3348    3351    3359    3368    3370    3372    3381    3391    3393    3394    3397
                3398    3399    3411    3420    3422    3423    3424    3425    3428    3438    3440    3446    3447    3450
                3454    3457    3460    3464    3466    3467    3468    3469    3472    3476    3477    3484    3487    3493
                3494    3500    3504    3506    3507    3508    3509    3512    3514    3515    3517    3527    3533    3534
                3540    3544    3546    3547    3548    3549    3552    3554    3555    3557    3564    3567    3573    3574    3580
                3584    3586    3587    3588    3589    3592    3594    3595    3597    3604    3607    3613    3614    3620    3624
                3627    3629    3634    3635    3638    3640    3642    3643    3646    3656    3665    3666    3675    3686    3691
                3708    3709    3710    3712    3716    3726    3727    3734    3735    3737    3746    3747    3749    3759    3760
                3769    3770    3776    3788    3796    3797    3799    3800    3801    3804    3805    3807    3809    3814    3815
                3820    3821    3822    3834    3842    3843    3846    3847    3852    3853    3855    3857    3858    3859    3862    3870
                3877    3886    3888    3890    3891    3892    3894    3895    3903    3904    3911    3913    3919    3929    3931
                3932    3935    3936    3937    3940    3967    3968    3985    3987    3990    3991    3994    3995    3996    3998
                4006    4008    4012    4016    4019    4028    4041    4043    4053    4055    4058    4059    4060    4062    4064
                4065    4067    4075    4080    4082    4083    4092    4093    4099    4110    4118    4120    4123    4124    4125
                4127    4129    4130    4138    4140    4145    4146    4147    4156    4157    4163    4174    4182    4184    4187
                4189    4190    4192    4193    4194    4198    4207    4209    4218    4219    4238    4240    4245    4246    4248
                4249    4250    4251    4254    4256    4264    4265    4267    4276    4280    4281    4288    4304    4305    4311
                4319    4325    4333    4334    4340    4353    4355    4358    4359    4363    4364    4365    4366    4367    4368
                4370    4379    4380    4382    4390    4396    4405    4407    4408    4414    4415    4416    4417    4419    4421
                4422    4424    4426    4427    4428    4429    4435    4445    4452    4456    4457    4460    4466    4467    4488
                4496    4497    4502    4510    4511    4515    4517    4519    4520    4527    4534    4535    4537    4539    4540
                4542    4552    4553    4554    4565    4571    4589    4594    4596    4597    4604    4610    4611    4615    4617
                4618    4622    4624    4626    4632    4639    4642    4651    4652    4654    4667    4668    4669    4689    4691
                4692    4693    4694    4696    4698    4697    4698    4706    4707    4709    4717    4721    4722    4725    4732
                4744    4750    4752    4753    4760    4767    4771    4772    4773    4774    4776    4785    4786    4788    4790
                4796    4800    4802    4803    4804    4805    4806    4808    4817    4818    4820    4826    4834    4835    4837
                4839    4845    4850    4856    4868    4870    4873    4876    4877    4878    4879    4881    4890    4891    4905
                4906    4917    4919    4922    4924    4925    4926    4927    4929    4937    4941    4949    4954    4961    4973
                4984    4985    4986    4987    4988    4989    4990    4991    4992    4993    4994    4995    5005    5018    5019
                5020    5021    5030    5031    5032    5033    5034    5035    5042    5043    5044    5053    5054    5055    5056
                5057    5073    5097    5098    5099    5101    5111    5132    5141    5144    5145    5146    5147    5148    5157
                5158    5170    5177    5178    5184    5187    5193    5201    5209    5211    5216    5218    5220    5238    5239
                5257    5258    5269    5270    5271    5272    5274    5276    5282    5283    5284    5295    5287    5289    5298
                5299    5302    5323    5325    5326    5328    5329    5330    5339    5347    5348    5355    5356    5364    5365
                5367    5368    5370    5371
MOVB            2596    2712    2722    2744    2770    2782    2819    2829    2853    2863    2924    2937    2964    2975    3039
                3051    3085    3097    3130    3141    3164    3175    3195    3198    3204    3205    3210    3211    3241    3244
                3250    3251    3256    3257    3286    3289    3295    3296    3301    3302    3331    3334    3340    3341    3346
                3347    3426    3429    3470    3473    3510    3513    3550    3553    3590    3593    3630    3758    3768    3784
                3802    3992    4282    4426    4430    4533    4619    4704    4769    4801    4908    4923    4939    4950    4985
```

DZDHMA.P11 CROSS REFERENCE TABLE

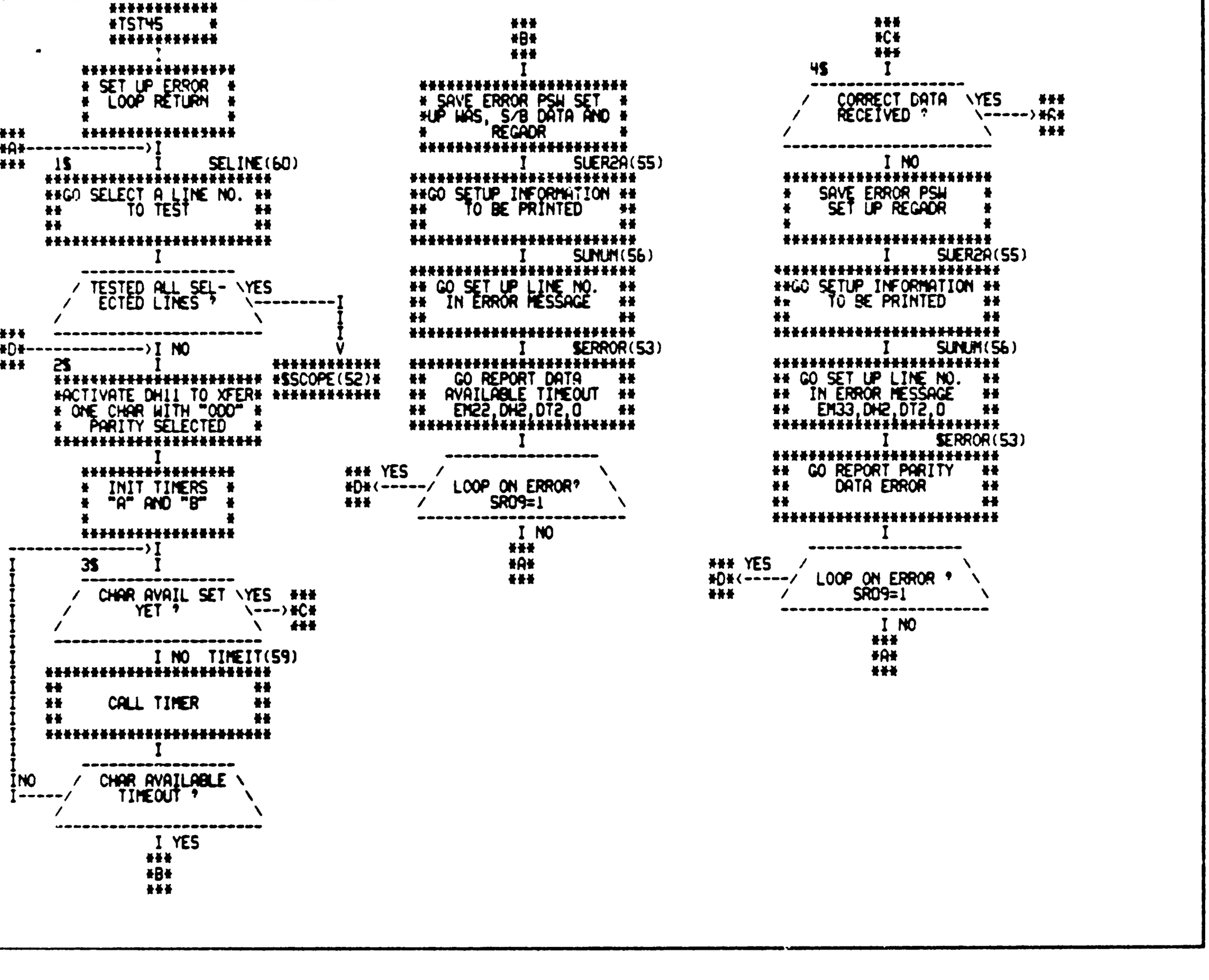
```
          4986  4988  4989  4990  4991  4992  4993  4994  5007  5017  5029  5051  5052  5100  5107
          5110  5118  5131  5150  5273  5286  5332
NEG       4988  4989
NOP       3443  3481  3521  3561  3601  3650  4349  4864  4984
RESET     4984  5278  5291
ROL       3980  4525  4602  4758  4988  4993
RTI       3451  3458  4484  4985  4986  4988  4989  4990  4992  4993  4995  5349  5357  5369
RTS       3722  3973  4987  4990  4991  4994  5011  5023  5037  5045  5058  5085  5112  5125  5137
          5162  5180  5246  5265  5305  5318  5341  5350  5358  5372
SEC       3977  4522  4599  4755
SUB       2654  3893  4410  4536  4616  4986  4989  4991
SWAB      3872  3914  3970  4076  4100  4143  4166  4221  4284  4297  4361  4458  4640  4723  4791
          4840  4951
TRAP      4994  5366
TST       2596  2603  2608  2612  2657  2681  3665  3739  3905  3947  3956  3975  4069  4307  4321
          4490  4504  4680  4810  4883  4931  4985  4986  4987  4988  4989  4990  4991  4993  4994
          5221
TSTB      3034  3046  3080  3092  3427  3471  3511  3551  3591  3945  4134  4200  4257  4372  4442
          4544  4683  4711  4901  4985  4989  4990  4991  4992  5070
.ASCII    2259
.ASCIZ    2259  4984  4987  4995  5623  5624  5630  5631  5637  5641  5645  5649  5653  5657  5661
          5665  5669  5673  5677  5681  5635  5686  5690  5634  5698  5702  5703  5707  5711  5715
          5719  5720  5726  5730  5731  5735  5736  5742  5746  5750  5751  5757  5761  5765  5769
          5773  5777  5781  5785  5789  5790  5796  5799  5803  5804  5808  5812  5816  5820  5824
          5828  5829  5830  5831  5832  5833  5834  5835  5836  5838
.BLKB     4992  5352
.BLKW     4989  5395  5499  5842  5846
.BYTE     2259  4984  4986  4988  4991  4992  5592  5593
.ENABL    2243
.END      5854
.ENDC     2244  2245  2247  2251  2252  2258  2259  2595  2596  2647  2671  2686  2690  2698  2707
          2732  2739  2755  2763  2805  2814  2839  2848  2873  2882  2910  2919  2948  2957  2986
          2995  3026  3033  3045  3072  3079  3091  3118  3123  3134  3152  3157  3168  3186  3192
          3232  3238  3278  3285  3323  3330  3368  3375  3420  3464  3504  3544  3584  3624  3717
          3725  3729  3797  3853  3886  3929  3985  3989  4053  4057  4118  4122  4182  4186  4238
          4242  4322  4353  4357  4405  4413  4438  4515  4531  4582  4587  4594  4608  4689  4719
          4740  4750  4764  4868  4872  4917  4921  4984  4985  4986  4987  4988  4989  4990  4991
          4992  4993  4994  4995  4998  5000  5375  5377  5617  5619
.EQUIV    2258
.EVEN     2259  4987  4991  4995  5625  5632  5721  5737  5752  5791  5825  5839  5851
.IF       2244  2245  2247  2251  2252  2258  2259  2595  2596  2647  2671  2686  2690  2698  2707
          2732  2739  2755  2763  2805  2814  2839  2848  2873  2882  2910  2919  2948  2957  2986
          2995  3026  3033  3045  3072  3079  3091  3118  3123  3134  3152  3157  3168  3186  3192
          3232  3238  3278  3285  3323  3330  3358  3375  3420  3464  3504  3544  3584  3624  3717
          3725  3729  3797  3853  3886  3929  3985  3989  4053  4057  4118  4122  4182  4186  4238
          4242  4322  4353  4357  4405  4413  4438  4515  4531  4582  4587  4594  4608  4689  4719
          4740  4750  4764  4868  4872  4917  4921  4984  4985  4986  4987  4988  4989  4990  4991
          4992  4993  4994  4995  4998  5000  5375  5377  5617  5619
.IFF      2245  2251  2252  2258  2259  2596  2647  2671  2686  2690  2698  2707  2732  2739  2755
          2763  2805  2814  2839  2848  2873  2882  2910  2919  2948  2957  2986  2995  3026  3033
          3045  3072  3079  3091  3118  3123  3134  3152  3157  3168  3186  3192  3232  3238  3278
          3285  3323  3330  3368  3375  3420  3464  3504  3544  3584  3624  3717  3725  3729  3797
          3853  3886  3929  3985  3989  4053  4057  4118  4122  4182  4186  4238  4242  4322  4353
          4357  4405  4413  4438  4515  4531  4582  4587  4594  4608  4689  4719  4740  4750  4764
```

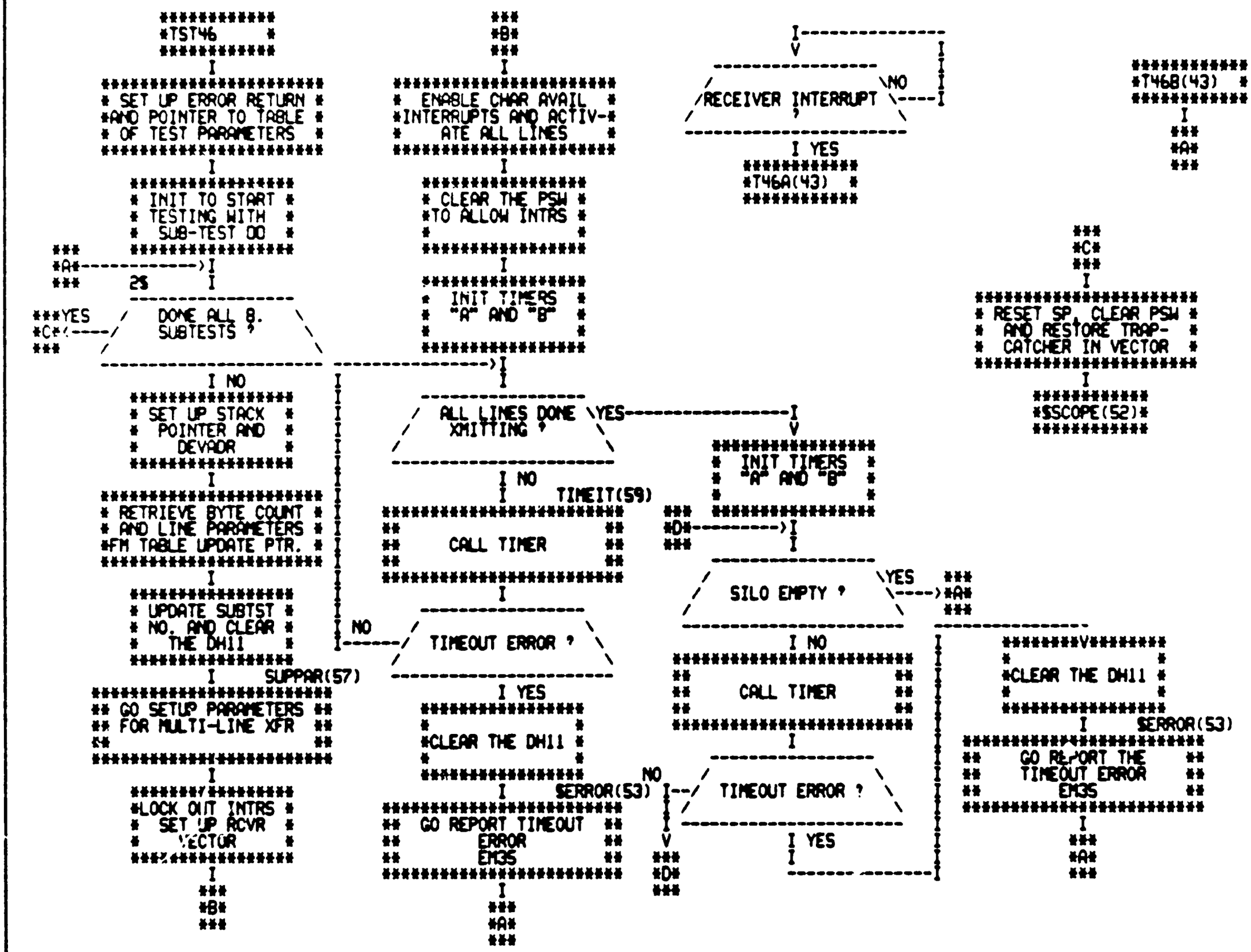
```
            4868    4872    4917    4921    4984    4985    4986    4987    4988    4989    4990    4991    4992    4993    4994
            4995    4998    5000    5375    5377    5617    5619
.IFT        4985    4986    4992    4993
.IFTF       4985    4986    4992    4993
.IIF        2244    2245    2246    2247    2259    2596    4984    4985    4986    4987    4990    4992    4993    4994
.IRP        2595    2647    2671    2698    2732    2755    2805    2839    2873    2910    2948    2986    3026    3072    3118
            3152    3186    3232    3278    3323    3368    3420    3464    3504    3544    3584    3624    3725    3797    3853
            3886    3929    3985    4053    4118    4182    4238    4353    4405    4515    4594    4689    4750    4868    4917
            4984    4989    4991    4993    4996    5097    5111
.LIST       2233    2236    2245    2247    2258    2259    2595    2596    2647    2671    2698    2732    2755    2805    2839
            2873    2910    2948    2986    3026    3072    3118    3152    3186    3232    3278    3323    3368    3420    3464
            3504    3544    3584    3624    3725    3797    3853    3886    3929    3985    4053    4118    4182    4238    4353
            4405    4515    4594    4689    4750    4868    4917    4984    4985    4986    4992    4994
.MACRO        1      45      93     149       1       [illegible]  297     350     405     448     491     534     633     732     773
            814     859     904     946     9[illegible]  1052    1100    1154    1199    1244    1287    1361    1432    1493    1537
            1572    1622    1675    1732    1763    1817    1863    1917    1950    2015    2047    2078    2136    2184    2245
            2263    2269    4994
.MCALL      2238    2239    2240    2[illegible]  2242    2258    2259    2596
.NLIST      2233    2235    2245    2[illegible]  2[illegible]  2[illegible]  2595    2596    2647    2671    2698    2732    2755    2805    2839
            2873    2910    2948    2[illegible]  3[illegible]  3072    3118    3152    3186    3232    3278    3323    3368    3420    3464
            3504    3544    3584    3[illegible]  3[illegible]  3797    3853    3[illegible]  3929    3985    4053    4118    4182    4238    4353
            4405    4515    4594    4[illegible]  4750    4868    4917    4[illegible]  4985    4986    4992    4994
.PAGE       2259    2582    2598    2[illegible]  2670    2697    2731    2754    2804    2838    2872    2909    2947    2985    3025
            3071    3117    3151    3[illegible]  3231    3277    3322    3367    3419    3463    3503    3543    3583    3623    3724
            3796    3852    3885    3[illegible]  3984    4052    4117    4181    4237    4352    4404    4514    4593    4688    4749
            4867    4916    4971    4[illegible]  5616
.REM        2648    2672    2699    2733    2756    2806    2840    2874    2911    2949    2987    3027    3073    3119    3153
            3187    3233    3279    3324    3369    3421    3465    3505    3545    3585    3625    3726    3798    3854    3887
            3930    3986    4054    4119    4183    4239    4354    4406    4516    4595    4690    4751    4869    4918
.REPT       2247    2259
.SBTTL      2245    2247    2251    2252    2258    2259    2647    2671    2698    2732    2755    2805    2839    2873    2910
            2948    2986    3026    3072    3118    3152    3186    3232    3278    3323    3368    3420    3464    3504    3544
            3584    3624    3725    3797    3853    3886    3929    3985    4053    4118    4182    4238    4353    4405    4515
            4594    4689    4750    4868    4917    4984    4985    4986    4987    4988    4989    4990    4991    4992    4993
            4994    4995
.TITLE      2244
.WORD       2247    2251    2252    2259    4984    4987    4988    4990    4991    4993    4995    5626    5633    5722    5738
            5753    5792


ERRORS DETECTED:  0

*DZDHMA,DZDHMA/CRF=DHMMAC,DZDHMA
RUN-TIME:  40 30  4 SECONDS
CORE USED:  37K
```

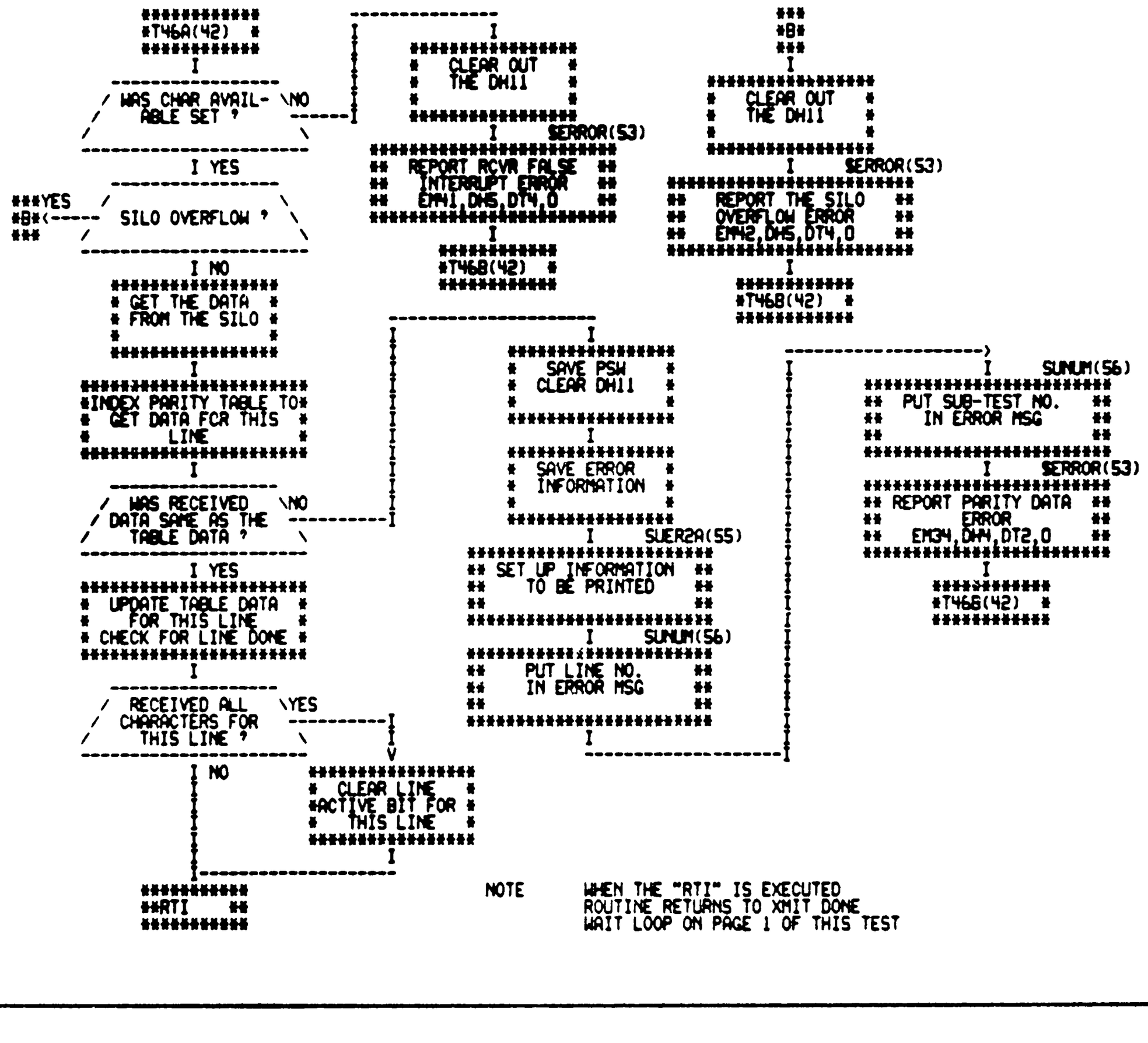
Spooler runtime 35 Seconds, 216 KCS, 1371 disk reads, 1 disk writes, 330 pages printed

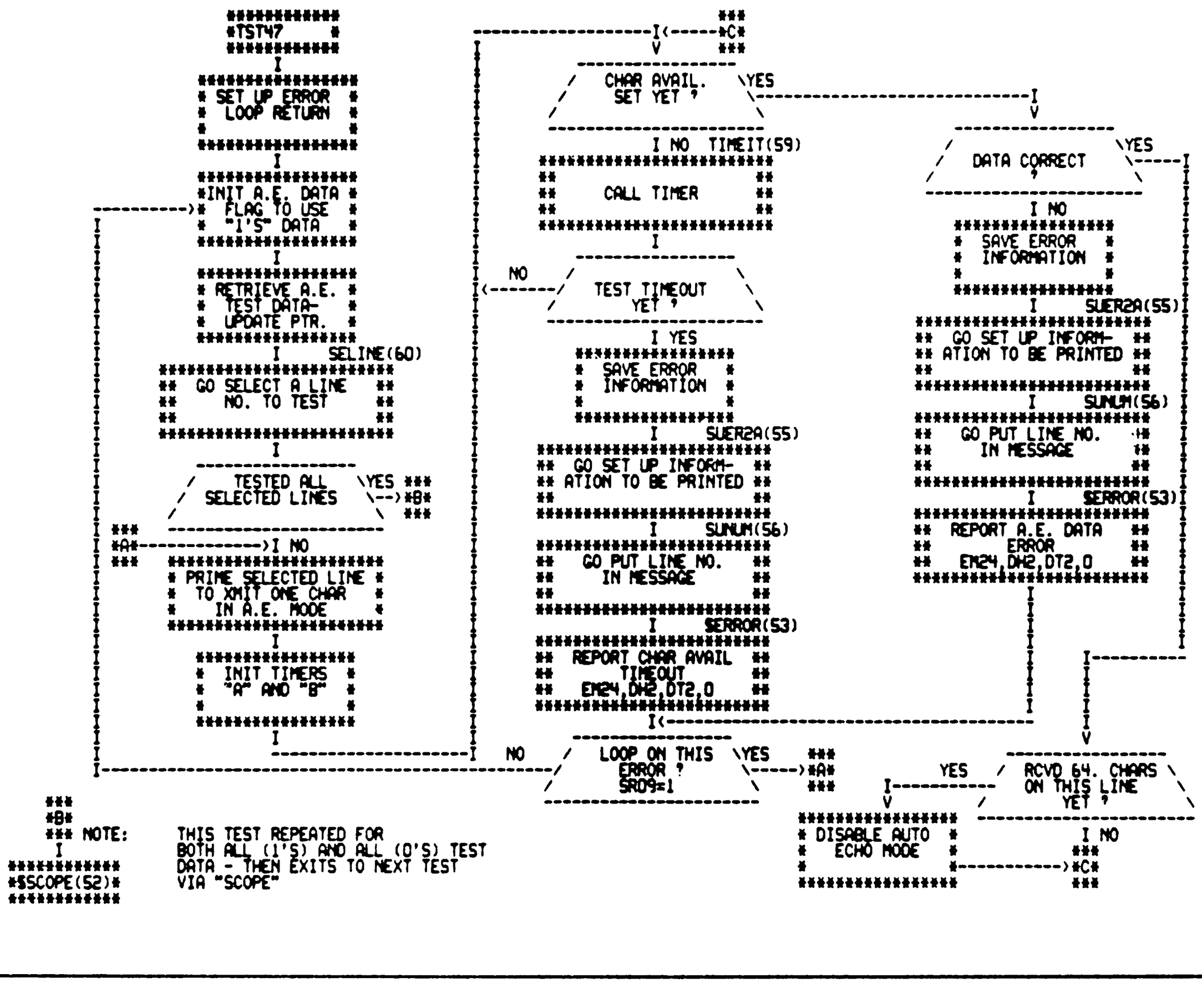


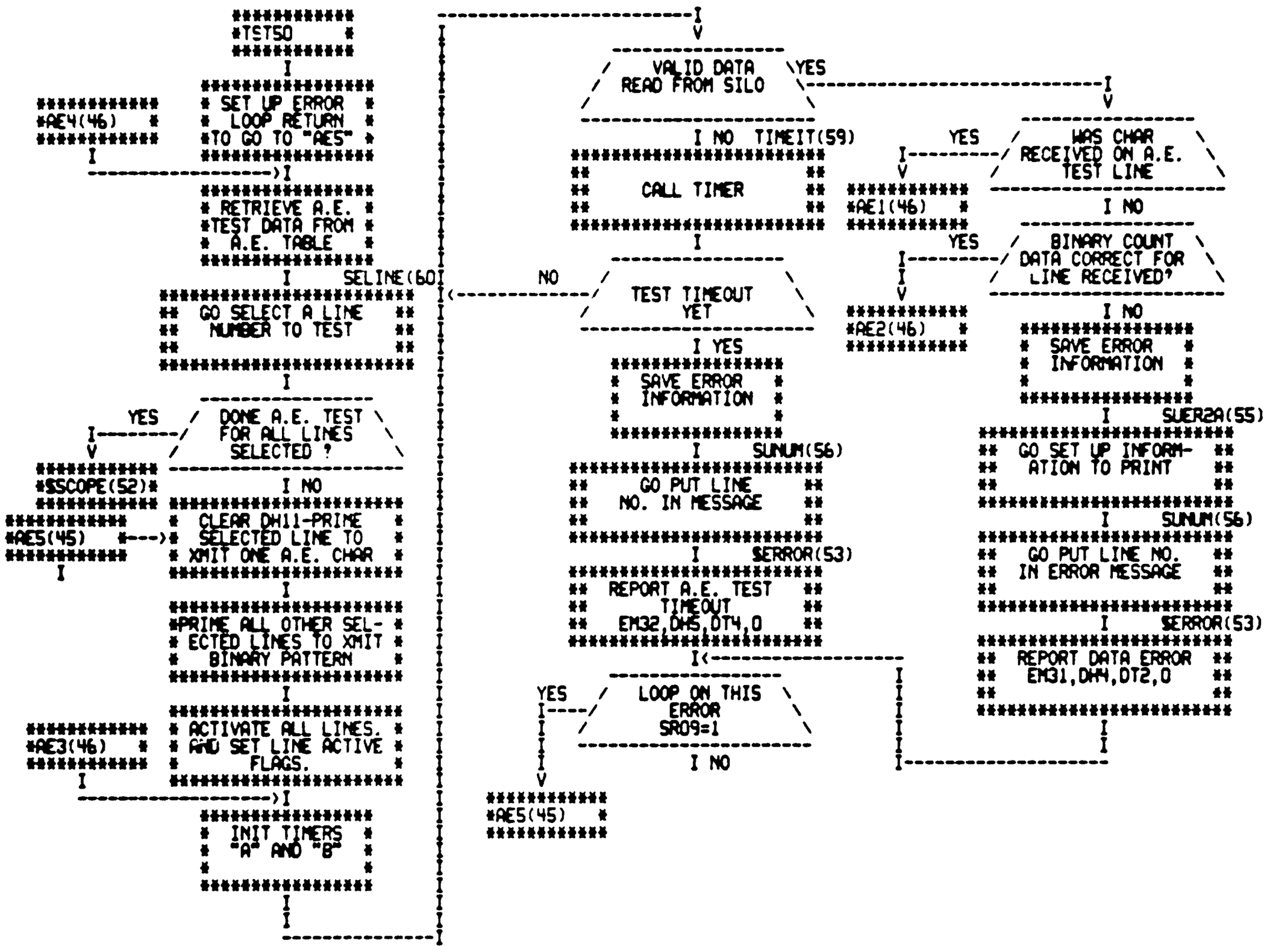
TEST 45 BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY

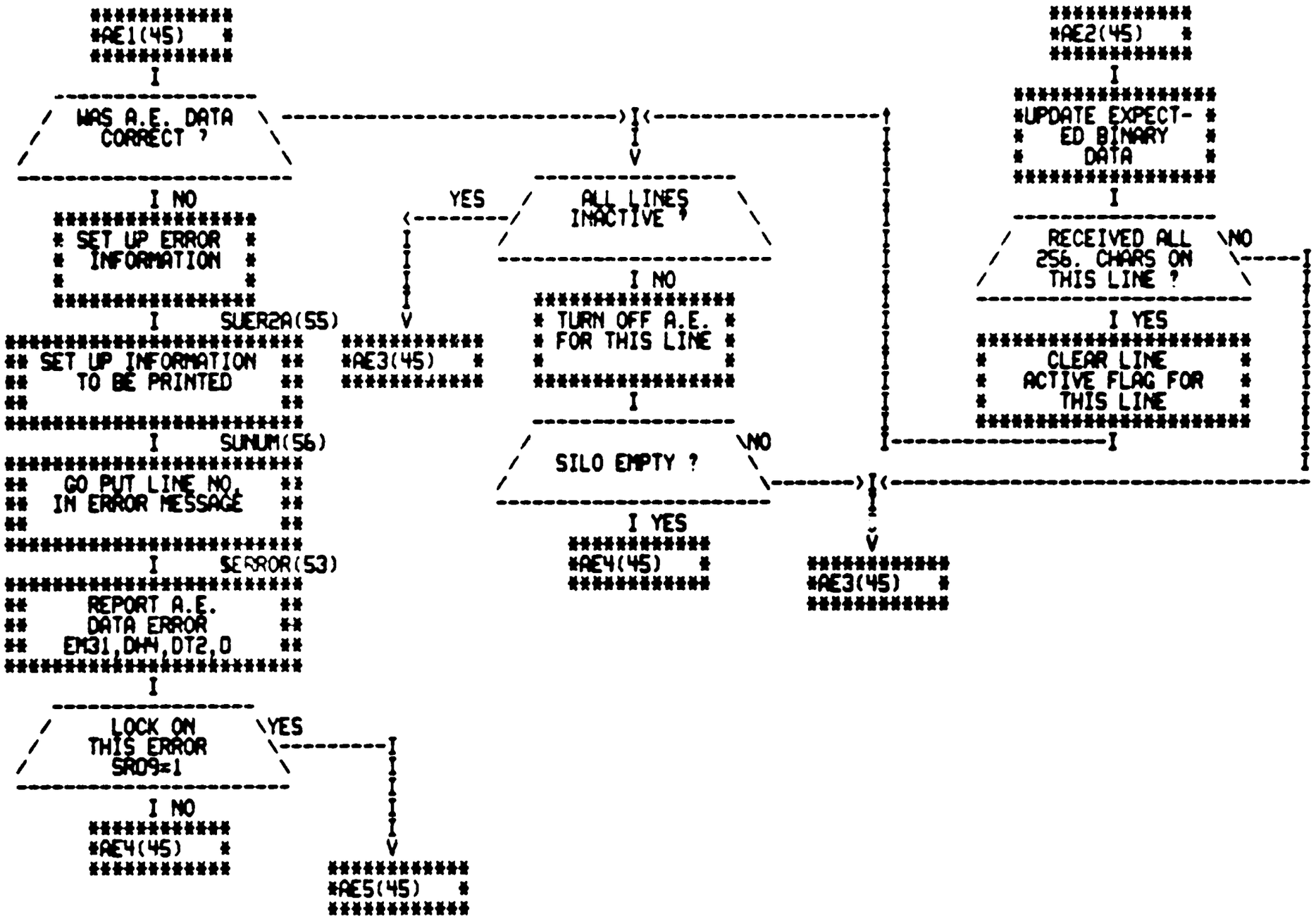


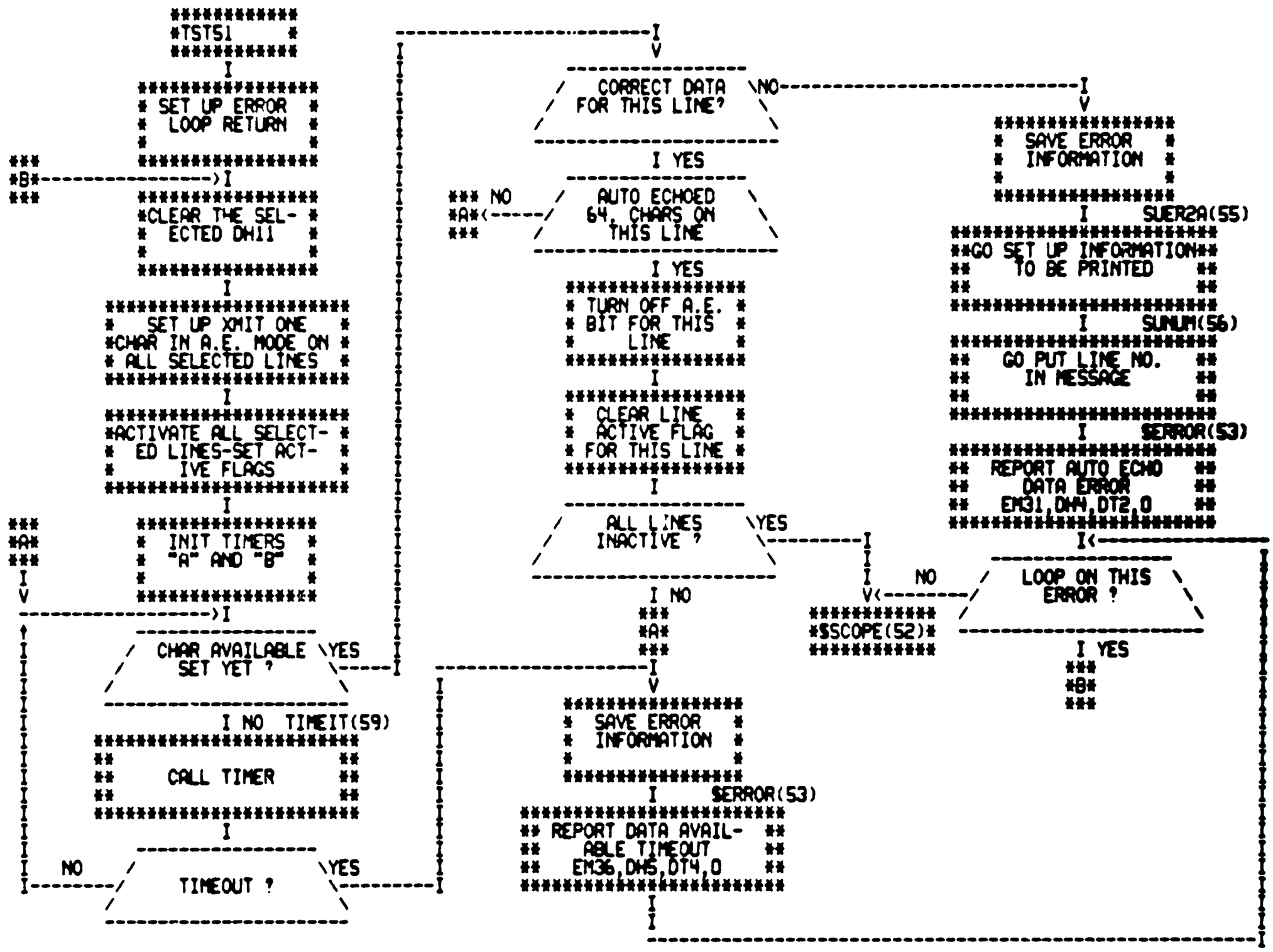

TST46
B
SET UP ERROR RETURN AND POINTER TO TABLE OF TEST PARAMETERS
INIT TO START TESTING WITH SUB-TEST 00
A
25
DONE ALL 8. SUBTESTS ?
YES
C
NO
SET UP STACK POINTER AND DEVADR
RETRIEVE BYTE COUNT AND LINE PARAMETERS #FM TABLE UPDATE PTR.
UPDATE SUBTST NO. AND CLEAR THE DH11
SUPPAR(57)
GO SETUP PARAMETERS FOR MULTI-LINE XFR
LOCK OUT INTRS SET UP RCVR VECTOR
B
ENABLE CHAR AVAIL INTERRUPTS AND ACTIV-ATE ALL LINES
CLEAR THE PSW TO ALLOW INTRS
INIT TIMERS "A" AND "B"
ALL LINES DONE XMITTING ?
YES
NO
TIMEIT(59)
CALL TIMER
TIMEOUT ERROR ?
YES
NO
CLEAR THE DH11
SERROR(53)
GO REPORT TIMEOUT ERROR EM35
A
RECEIVER INTERRUPT
NO
YES
T46A(43)
INIT TIMERS "A" AND "B"
D
SILO EMPTY ?
YES
NO
CALL TIMER
TIMEOUT ERROR ?
YES
NO
CLEAR THE DH11
SERROR(53)
GO REPORT THE TIMEOUT ERROR EM35
A
T46B(43)
A
C
RESET SP, CLEAR PSW AND RESTORE TRAP-CATCHER IN VECTOR
$SCOPE(52)

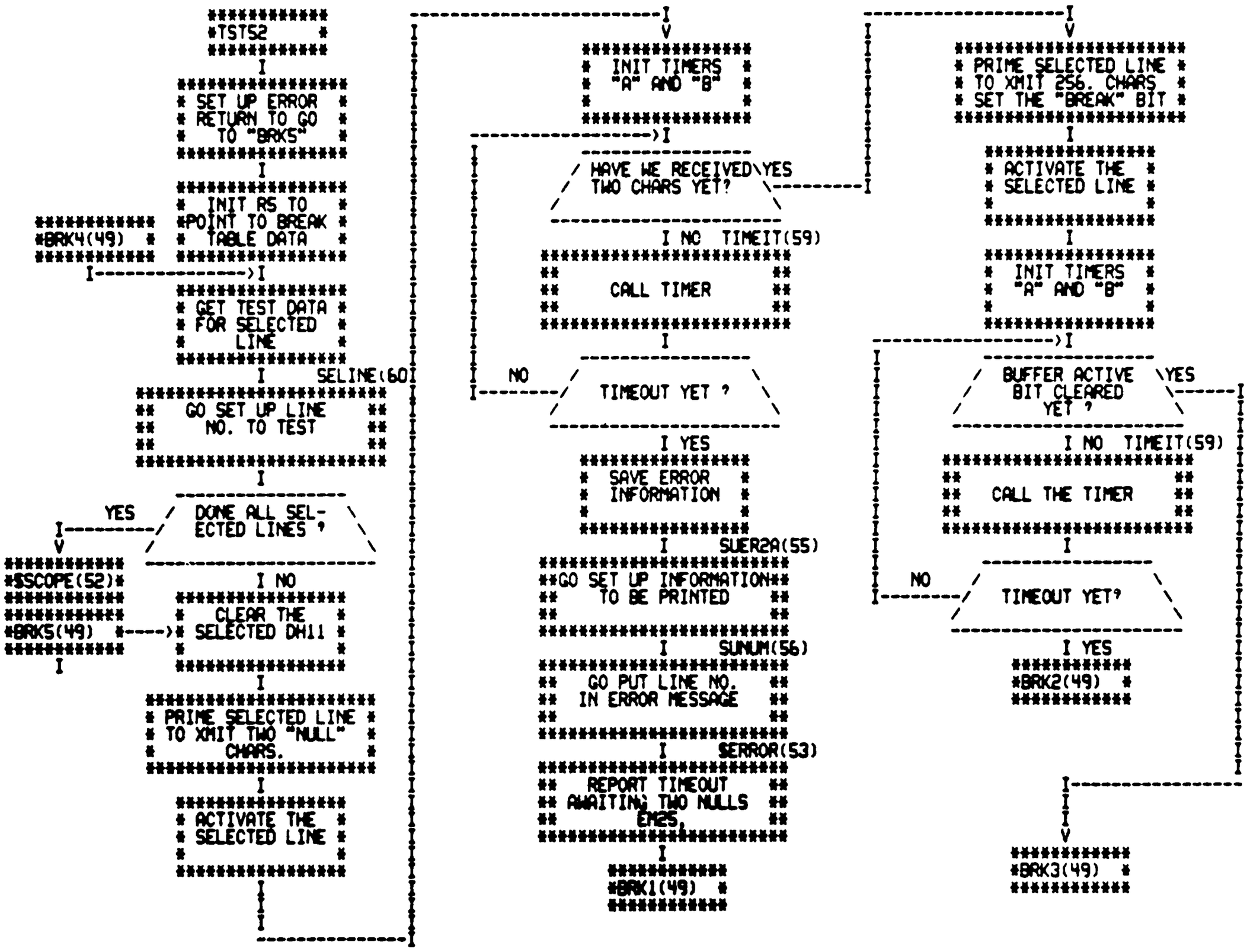

*T46A(42)
WAS CHAR AVAIL- ABLE SET ?
NO
YES
CLEAR OUT THE DH11
SERROR(53)
REPORT RCVR FALSE INTERRUPT ERROR EM41,DH5,DT4,0
*T46B(42)
***YES
8
***
SILO OVERFLOW ?
NO
CLEAR OUT THE DH11
SERROR(53)
REPORT THE SILO OVERFLOW ERROR EM42,DH5,DT4,0
*T46B(42)
GET THE DATA FROM THE SILO
INDEX PARITY TABLE TO GET DATA FOR THIS LINE
WAS RECEIVED DATA SAME AS THE TABLE DATA ?
NO
YES
SAVE PSW CLEAR DH11
SAVE ERROR INFORMATION
SUER2A(55)
SET UP INFORMATION TO BE PRINTED
SUNUM(56)
PUT LINE NO. IN ERROR MSG
SUNUM(56)
PUT SUB-TEST NO. IN ERROR MSG
SERROR(53)
REPORT PARITY DATA ERROR EM34,DH4,DT2,0
*T46B(42)
UPDATE TABLE DATA FOR THIS LINE CHECK FOR LINE DONE
RECEIVED ALL CHARACTERS FOR THIS LINE ?
NO
YES
CLEAR LINE ACTIVE BIT FOR THIS LINE
**RTI **
NOTE
WHEN THE "RTI" IS EXECUTED
ROUTINE RETURNS TO XMIT DONE
WAIT LOOP ON PAGE 1 OF THIS TEST

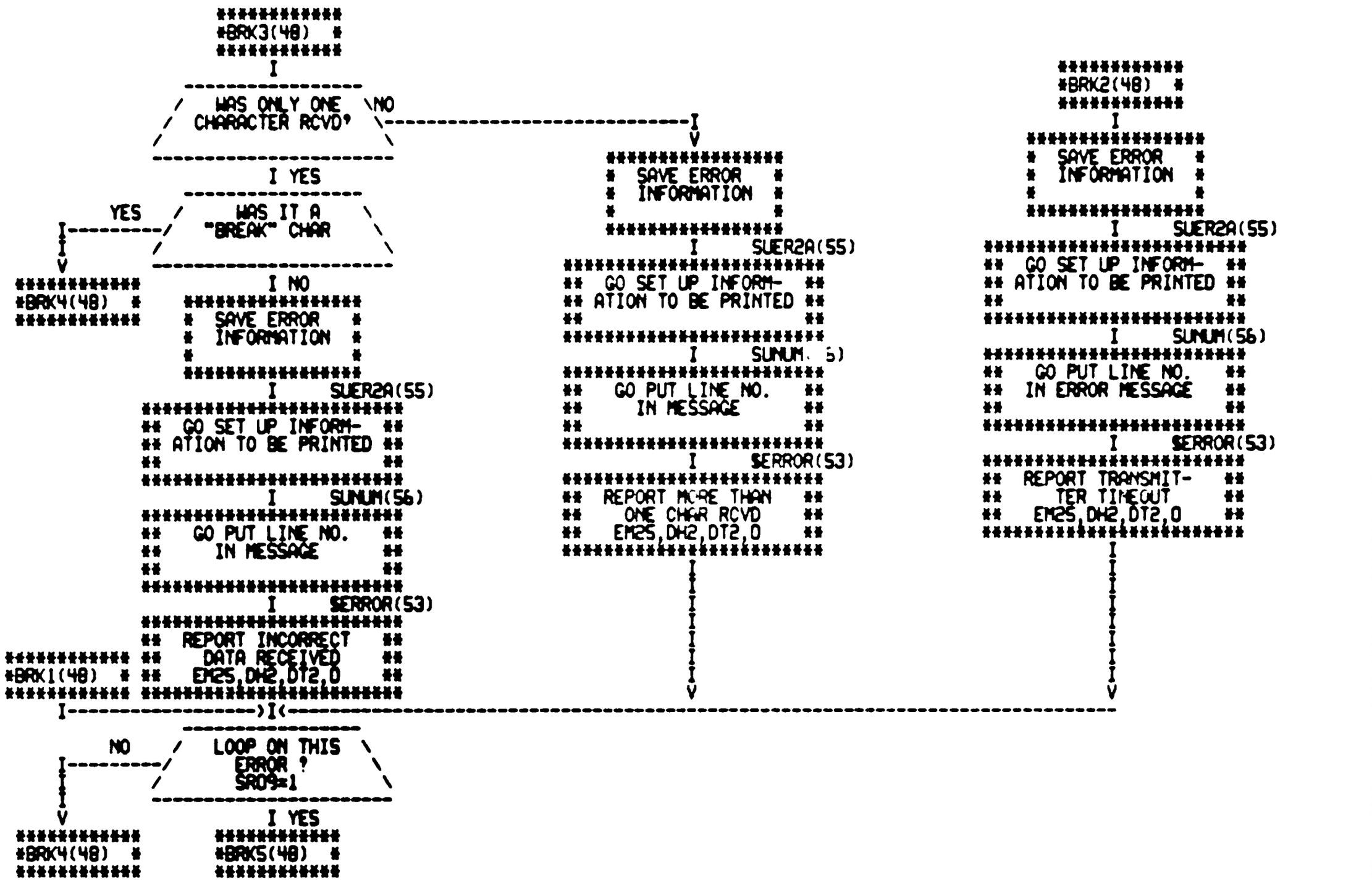

TST47
SET UP ERROR LOOP RETURN
INIT A.E. DATA FLAG TO USE "1'S" DATA
RETRIEVE A.E. TEST DATA- UPDATE PTR.
SELINE(60)
GO SELECT A LINE NO. TO TEST
TESTED ALL SELECTED LINES
YES
NO
PRIME SELECTED LINE TO XMIT ONE CHAR IN A.E. MODE
INIT TIMERS "A" AND "B"
LOOP ON THIS ERROR ? SR09=1
YES
NO
CHAR AVAIL. SET YET ?
YES
NO
TIMEIT(59)
CALL TIMER
TEST TIMEOUT YET ?
YES
NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT CHAR AVAIL TIMEOUT EM24,DH2,DT2,0
DATA CORRECT ?
YES
NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT A.E. DATA ERROR EM24,DH2,DT2,0
RCVD 64. CHARS ON THIS LINE YET ?
YES
NO
DISABLE AUTO ECHO MODE
#8
#SCOPE(52)
NOTE: THIS TEST REPEATED FOR BOTH ALL (1'S) AND ALL (0'S) TEST DATA - THEN EXITS TO NEXT TEST VIA "SCOPE"


*TST50
SET UP ERROR LOOP RETURN TO GO TO "AE5"
*AE4(46)
RETRIEVE A.E. TEST DATA FROM A.E. TABLE
SELINE(60)
GO SELECT A LINE NUMBER TO TEST
DONE A.E. TEST FOR ALL LINES SELECTED ?
YES
NO
*SSCOPE(52)
*AE5(45)
CLEAR DH11-PRIME SELECTED LINE TO XMIT ONE A.E. CHAR
PRIME ALL OTHER SELECTED LINES TO XMIT BINARY PATTERN
*AE3(46)
ACTIVATE ALL LINES. AND SET LINE ACTIVE FLAGS.
INIT TIMERS "A" AND "B"
VALID DATA READ FROM SILO
YES
NO TIMEIT(59)
CALL TIMER
TEST TIMEOUT YET
NO
YES
SAVE ERROR INFORMATION
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT A.E. TEST TIMEOUT EM32,DH5,DT4,0
LOOP ON THIS ERROR SR09=1
NO
YES
*AE5(45)
WAS CHAR RECEIVED ON A.E. TEST LINE
YES
NO
*AE1(46)
BINARY COUNT DATA CORRECT FOR LINE RECEIVED?
YES
NO
*AE2(46)
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM-ATION TO PRINT
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
SERROR(53)
REPORT DATA ERROR EM31,DH4,DT2,0

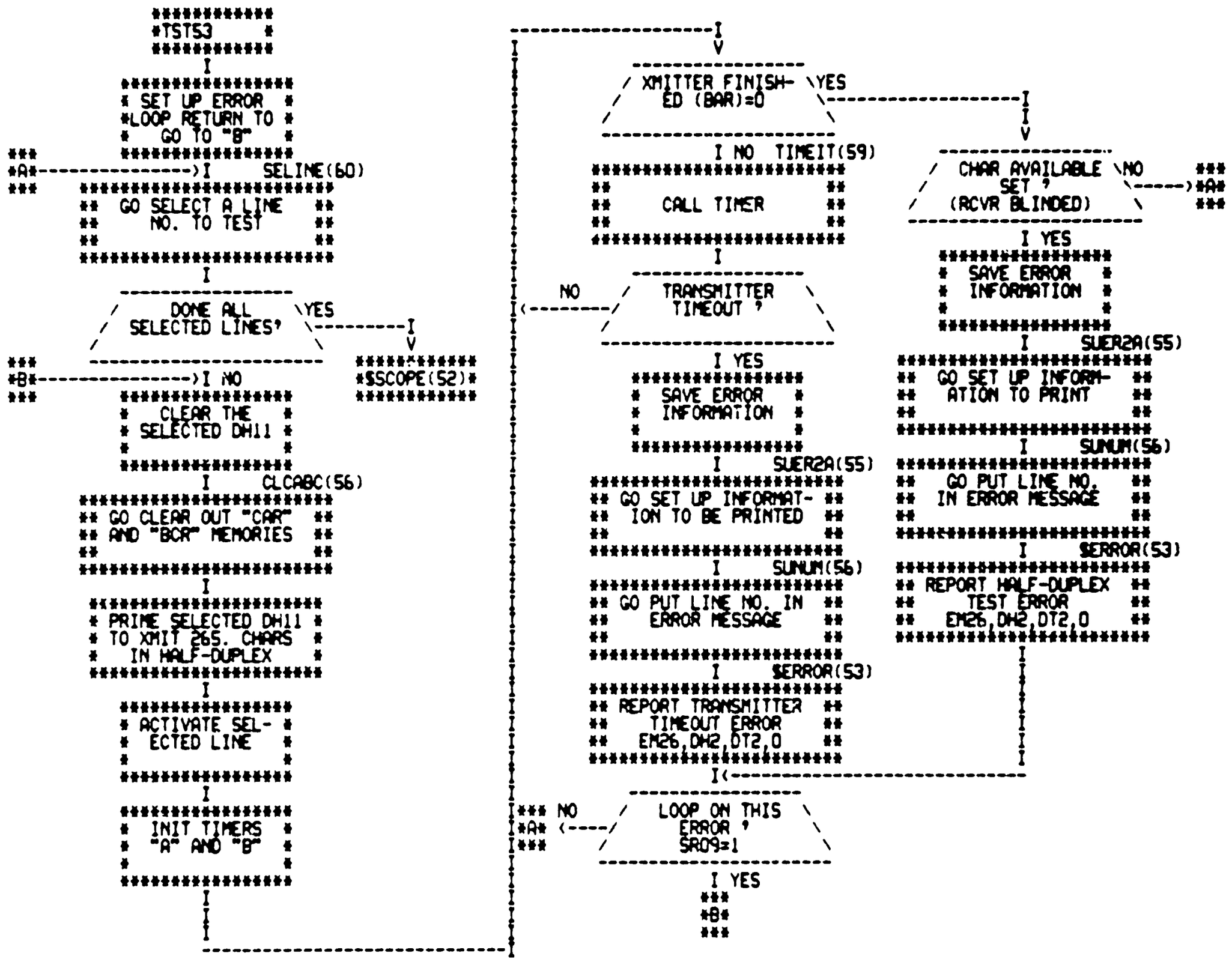

*AE1(45) *
WAS A.E. DATA CORRECT ?
I NO
* SET UP ERROR * * INFORMATION *
SUER2A(55)
** SET UP INFORMATION ** TO BE PRINTED **
SUNUM(56)
** GO PUT LINE NO. ** IN ERROR MESSAGE **
SERROR(53)
** REPORT A.E. ** DATA ERROR ** EM31,DHM,DT2,0 **
LOCK ON THIS ERROR SR09=1
YES
I NO
*AE4(45) *
YES
*AE3(45) *
ALL LINES INACTIVE ?
I NO
* TURN OFF A.E. * * FOR THIS LINE *
SILO EMPTY ?
NO
I YES
*AE4(45) *
*AE5(45) *
*AE2(45) *
*UPDATE EXPECT- * * ED BINARY * * DATA *
RECEIVED ALL 256. CHARS ON THIS LINE ?
NO
I YES
* CLEAR LINE * * ACTIVE FLAG FOR * * THIS LINE *
*AE3(45) *

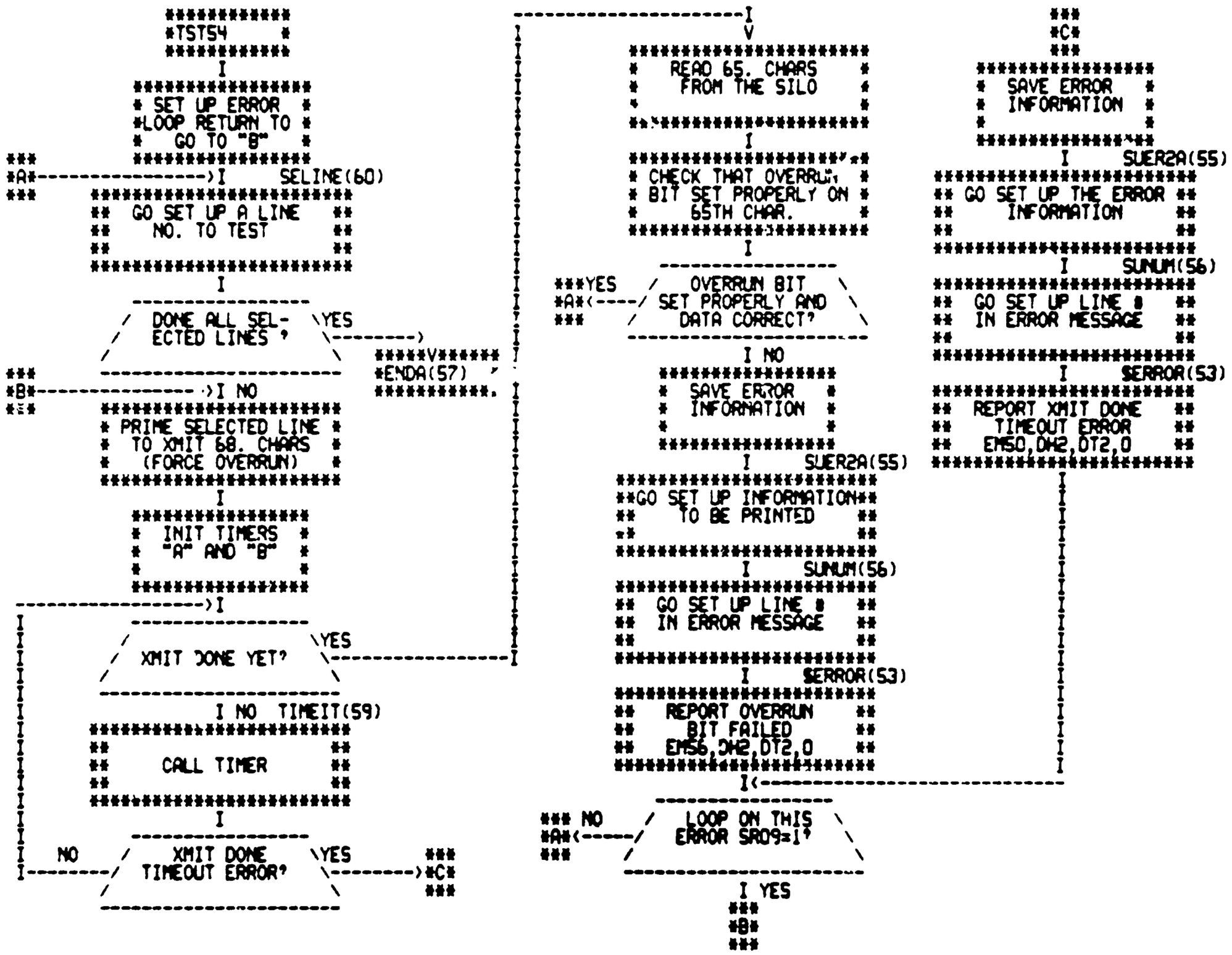

TSTS1
SET UP ERROR LOOP RETURN
B
CLEAR THE SEL- ECTED DH11
SET UP XMIT ONE CHAR IN A.E. MODE ON ALL SELECTED LINES
ACTIVATE ALL SELECT- ED LINES-SET ACT- IVE FLAGS
A
INIT TIMERS "A" AND "B"
CHAR AVAILABLE SET YET ?
YES
NO
CALL TIMER
TIMEIT(59)
TIMEOUT ?
YES
NO
A
CORRECT DATA FOR THIS LINE?
NO
YES
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT AUTO ECHO DATA ERROR EM31,DH4,DT2,0
LOOP ON THIS ERROR ?
YES
B
$SCOPE(52)
A
AUTO ECHOED 64. CHARS ON THIS LINE
NO
YES
TURN OFF A.E. BIT FOR THIS LINE
CLEAR LINE ACTIVE FLAG FOR THIS LINE
ALL LINES INACTIVE ?
YES
NO
A
SAVE ERROR INFORMATION
SERROR(53)
REPORT DATA AVAIL- ABLE TIMEOUT EM36,DH5,DT4,0

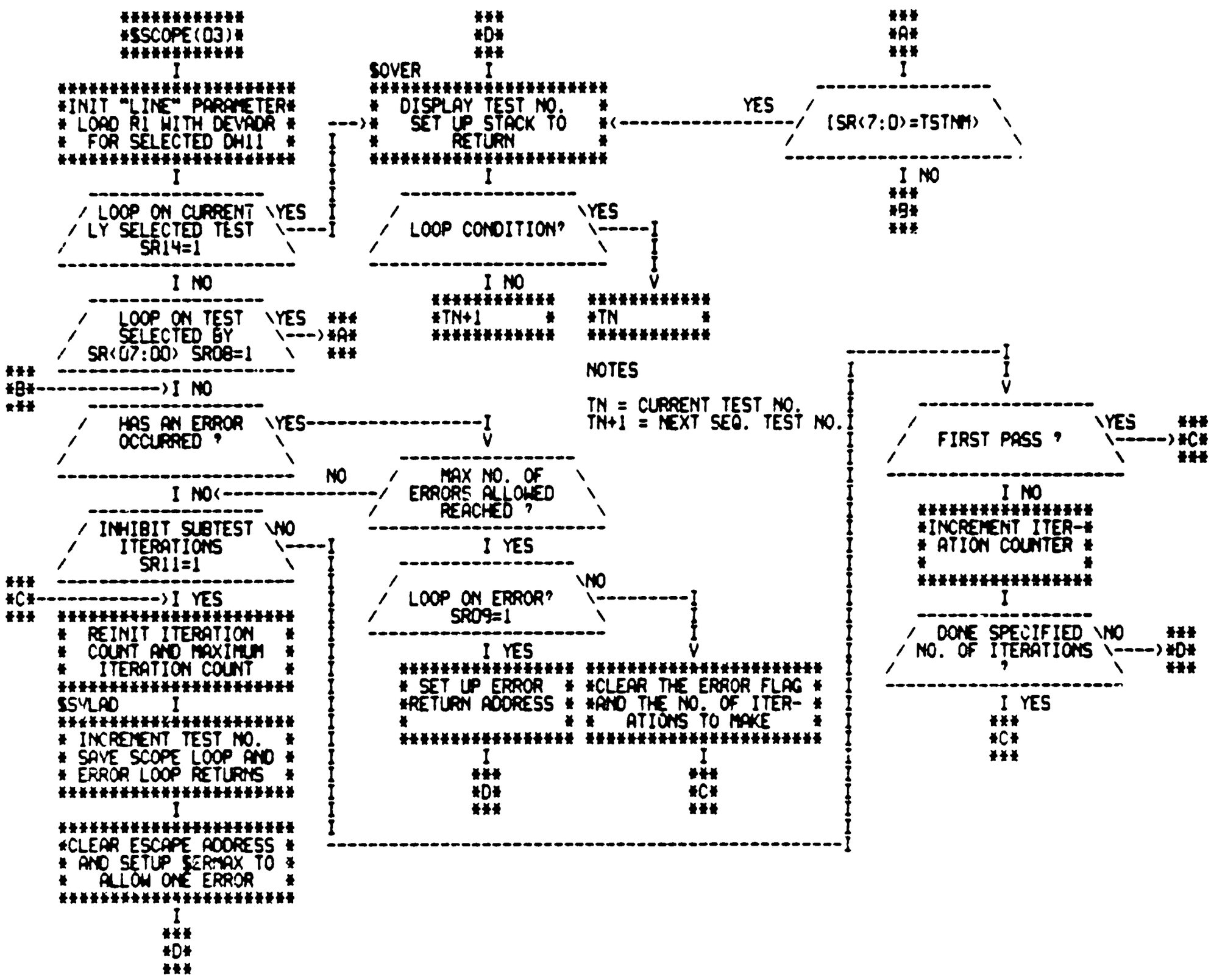

*TST52
SET UP ERROR RETURN TO GO TO "BRK5"
INIT RS TO POINT TO BREAK TABLE DATA
*BRK4(49)
GET TEST DATA FOR SELECTED LINE
SELINE(60)
GO SET UP LINE NO. TO TEST
DONE ALL SELECTED LINES ?
YES
NO
*SSCOPE(52)
*BRK5(49)
CLEAR THE SELECTED DH11
PRIME SELECTED LINE TO XMIT TWO "NULL" CHARS.
ACTIVATE THE SELECTED LINE
INIT TIMERS "A" AND "B"
HAVE WE RECEIVED TWO CHARS YET?
NO TIMEIT(59)
CALL TIMER
TIMEOUT YET ?
YES
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
SERROR(53)
REPORT TIMEOUT AWAITING TWO NULLS EM25,
*BRK1(49)
PRIME SELECTED LINE TO XMIT 256. CHARS SET THE "BREAK" BIT
ACTIVATE THE SELECTED LINE
INIT TIMERS "A" AND "B"
BUFFER ACTIVE BIT CLEARED YET ?
NO TIMEIT(59)
CALL THE TIMER
TIMEOUT YET?
YES
*BRK2(49)
*BRK3(49)

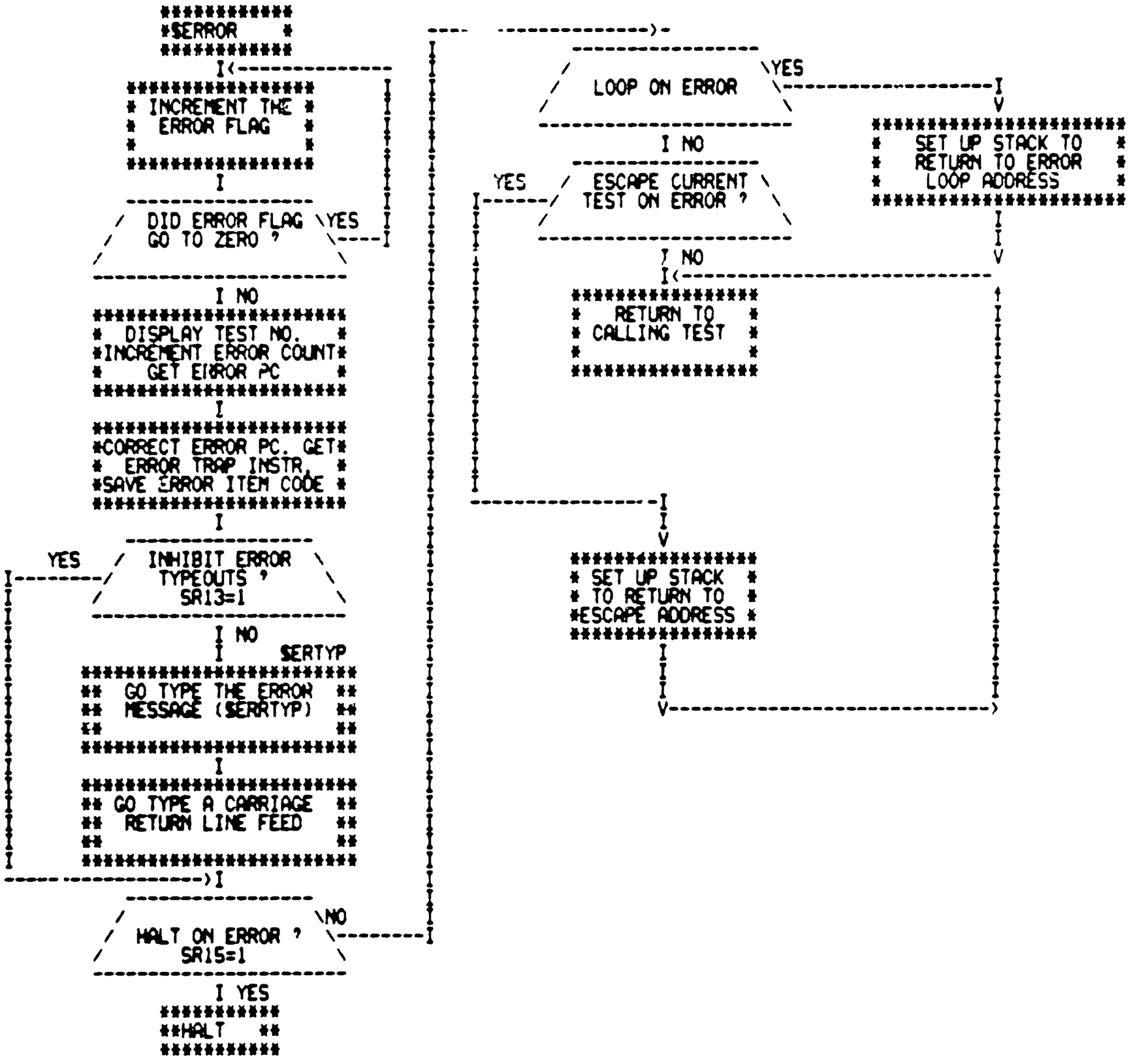

*BRK3(48) *
WAS ONLY ONE CHARACTER RCVD?
NO
YES
WAS IT A "BREAK" CHAR
YES
NO
*BRK4(48) *
*BRK1(48) *
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT INCORRECT DATA RECEIVED
EM25,DH2,DT2,0
*BRK2(48) *
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN ERROR MESSAGE
SERROR(53)
REPORT TRANSMIT- TER TIMEOUT
EM25,DH2,DT2,0
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORM- ATION TO BE PRINTED
SUNUM(56)
GO PUT LINE NO. IN MESSAGE
SERROR(53)
REPORT MORE THAN ONE CHAR RCVD
EM25,DH2,DT2,0
LOOP ON THIS ERROR ? SR09=1
NO
YES
*BRK4(48) *
*BRK5(48) *

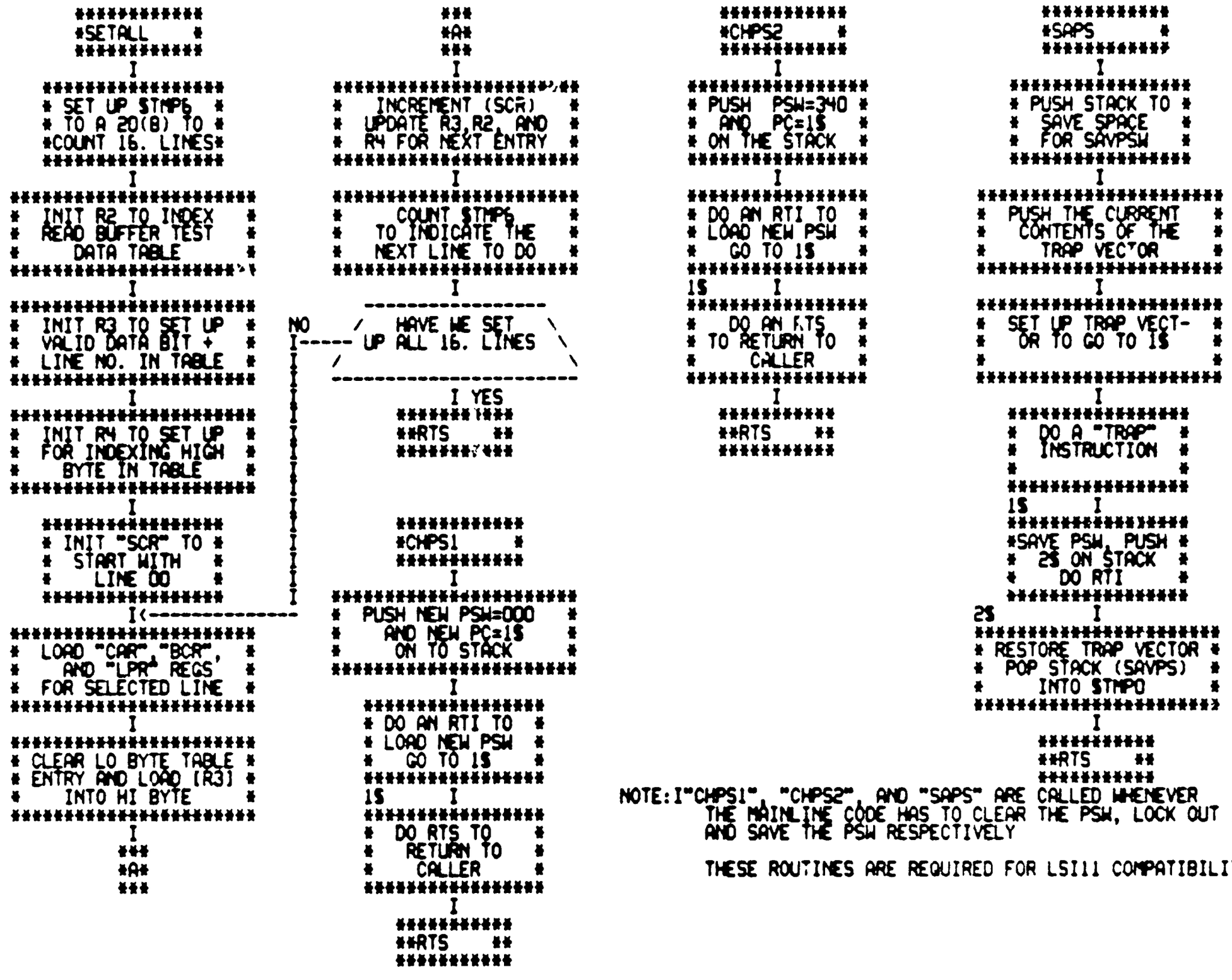
TEST 53 HALF-DUPLEX TEST-ALL LINES




TST54
SET UP ERROR LOOP RETURN TO GO TO "B"
SELINE(60)
GO SET UP A LINE NO. TO TEST
DONE ALL SELECTED LINES ?
YES
ENDA(57)
NO
PRIME SELECTED LINE TO XMIT 68. CHARS (FORCE OVERRUN)
INIT TIMERS "A" AND "B"
XMIT DONE YET?
YES
NO TIMEIT(59)
CALL TIMER
XMIT DONE TIMEOUT ERROR?
NO
YES
C
READ 65. CHARS FROM THE SILO
CHECK THAT OVERRUN BIT SET PROPERLY ON 65TH CHAR.
YES
OVERRUN BIT SET PROPERLY AND DATA CORRECT?
NO
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP INFORMATION TO BE PRINTED
SUNUM(56)
GO SET UP LINE # IN ERROR MESSAGE
SERROR(53)
REPORT OVERRUN BIT FAILED
EM56,DH2,DT2,0
LOOP ON THIS ERROR SR09=1?
NO
YES
B
C
SAVE ERROR INFORMATION
SUER2A(55)
GO SET UP THE ERROR INFORMATION
SUNUM(56)
GO SET UP LINE # IN ERROR MESSAGE
SERROR(53)
REPORT XMIT DONE TIMEOUT ERROR
EM50,DH2,DT2,0

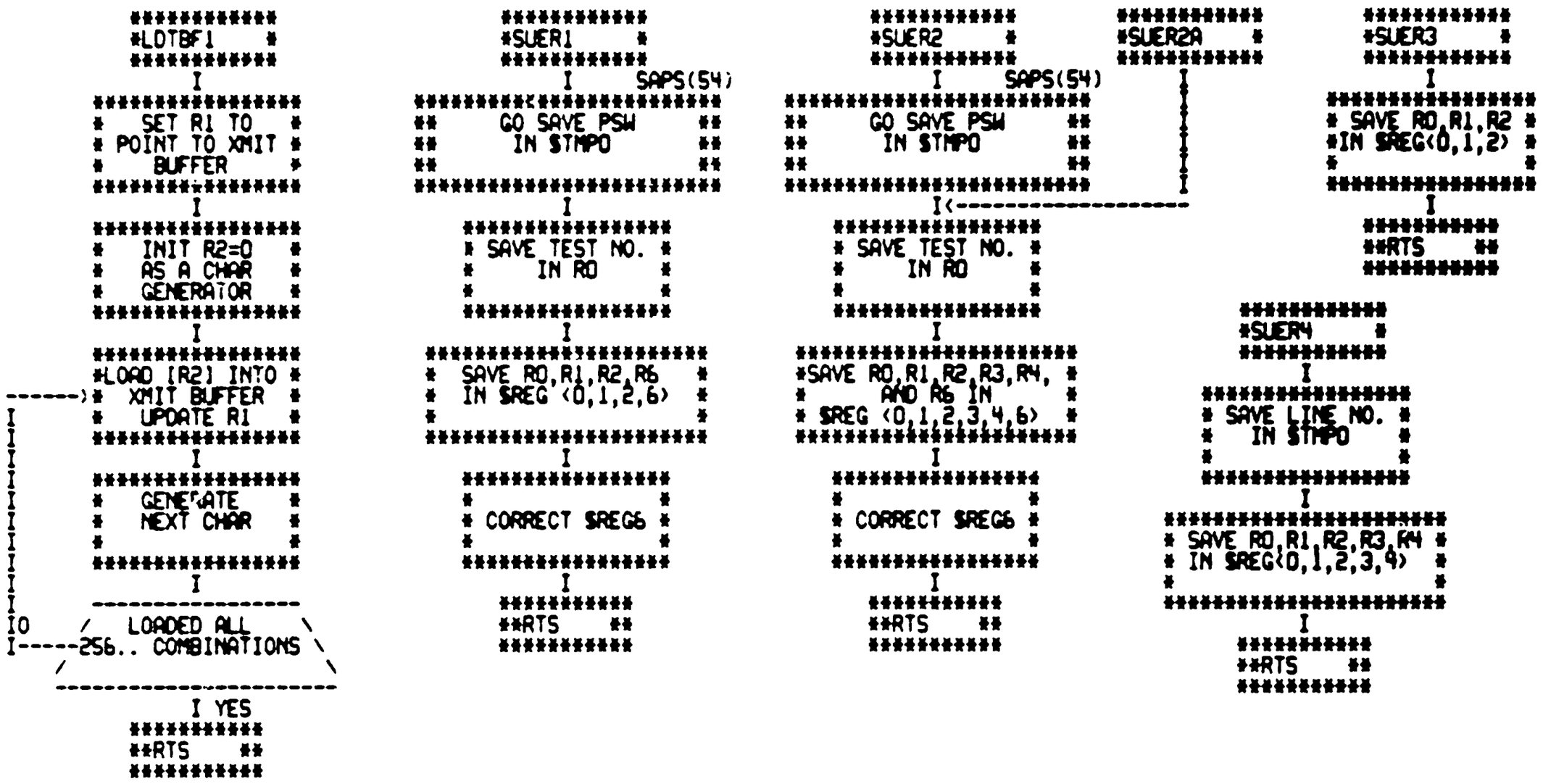

$SCOPE(03)
$OVER
INIT "LINE" PARAMETER
LOAD R1 WITH DEVADR FOR SELECTED DH11
DISPLAY TEST NO.
SET UP STACK TO RETURN
(SR<7:0>=TSTNM)
YES
NO
LOOP ON CURRENT LY SELECTED TEST SR14=1
LOOP CONDITION?
LOOP ON TEST SELECTED BY SR<07:00> SR08=1
HAS AN ERROR OCCURRED ?
MAX NO. OF ERRORS ALLOWED REACHED ?
INHIBIT SUBTEST ITERATIONS SR11=1
LOOP ON ERROR? SR09=1
REINIT ITERATION COUNT AND MAXIMUM ITERATION COUNT
INCREMENT TEST NO. SAVE SCOPE LOOP AND ERROR LOOP RETURNS
CLEAR ESCAPE ADDRESS AND SETUP $ERMAX TO ALLOW ONE ERROR
SET UP ERROR RETURN ADDRESS
CLEAR THE ERROR FLAG AND THE NO. OF ITERATIONS TO MAKE
TN
TN+1
NOTES
TN = CURRENT TEST NO.
TN+1 = NEXT SEQ. TEST NO.
FIRST PASS ?
INCREMENT ITERATION COUNTER
DONE SPECIFIED NO. OF ITERATIONS ?
A
B
C
D

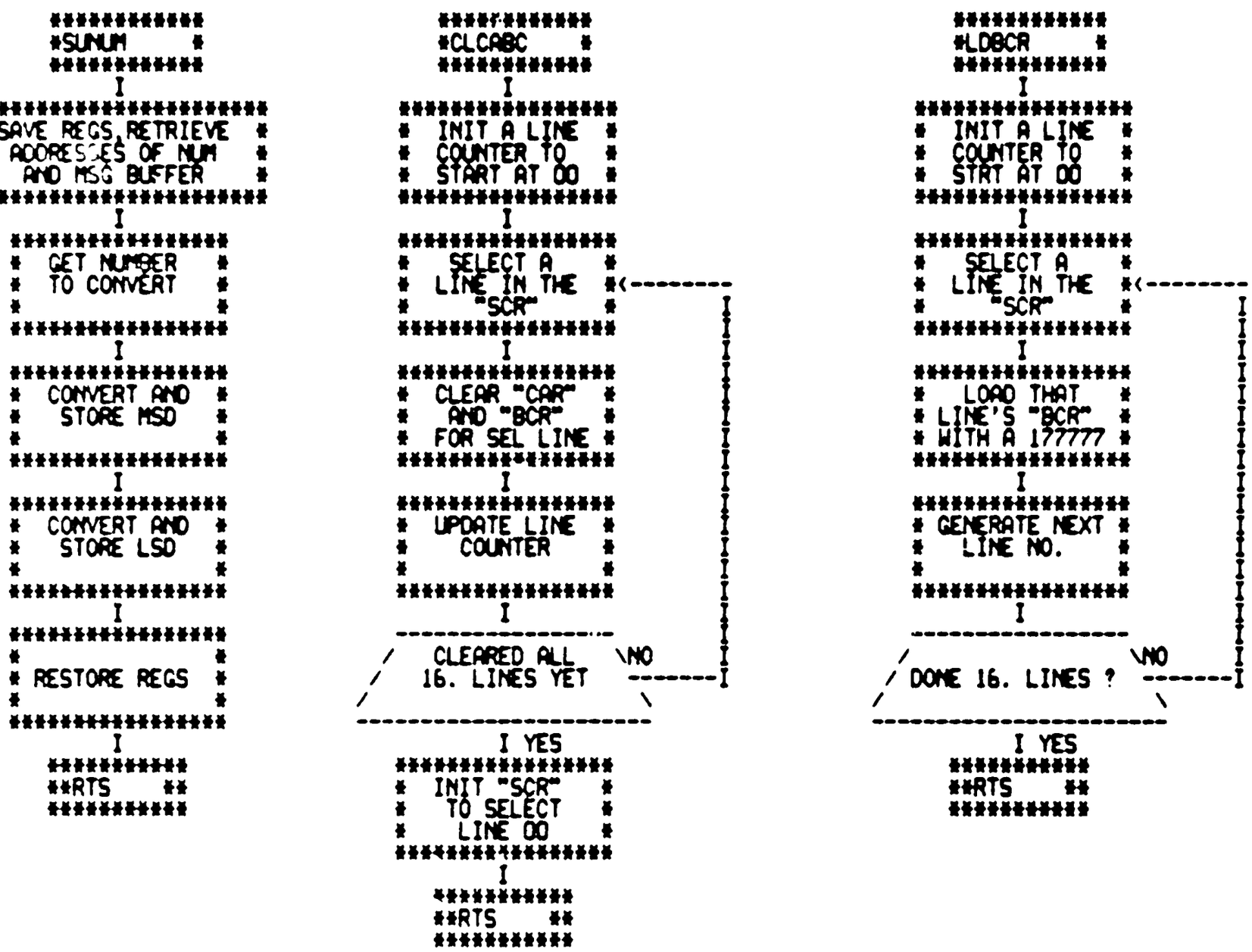

$ERROR
INCREMENT THE ERROR FLAG
DID ERROR FLAG GO TO ZERO ?
YES
NO
DISPLAY TEST NO. INCREMENT ERROR COUNT GET ERROR PC
CORRECT ERROR PC. GET ERROR TRAP INSTR. SAVE ERROR ITEM CODE
INHIBIT ERROR TYPEOUTS ? SR13=1
NO
$ERTYP
GO TYPE THE ERROR MESSAGE ($ERRTYP)
GO TYPE A CARRIAGE RETURN LINE FEED
HALT ON ERROR ? SR15=1
YES
HALT
LOOP ON ERROR
YES
NO
ESCAPE CURRENT TEST ON ERROR ?
YES
NO
SET UP STACK TO RETURN TO ERROR LOOP ADDRESS
RETURN TO CALLING TEST
SET UP STACK TO RETURN TO ESCAPE ADDRESS

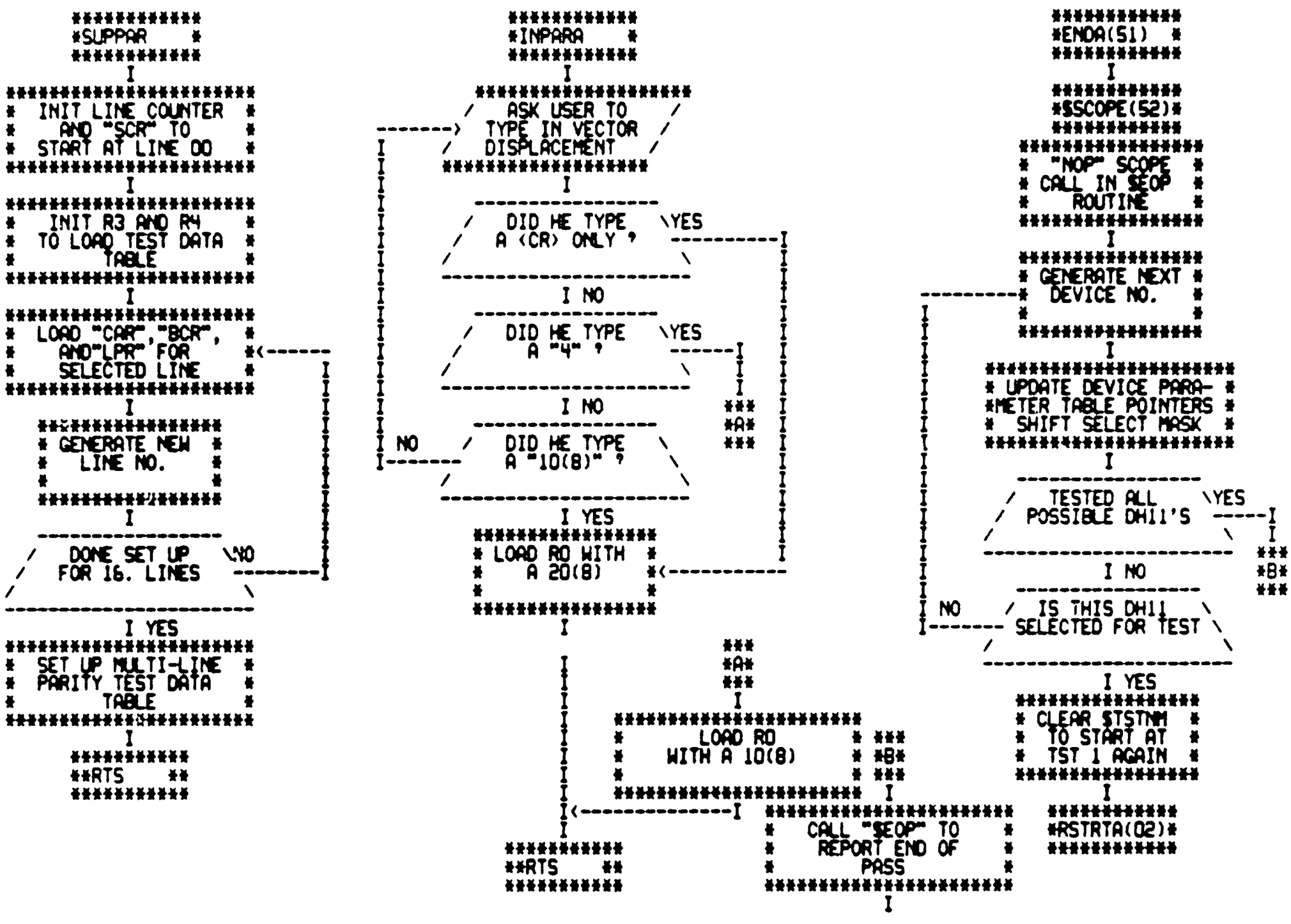


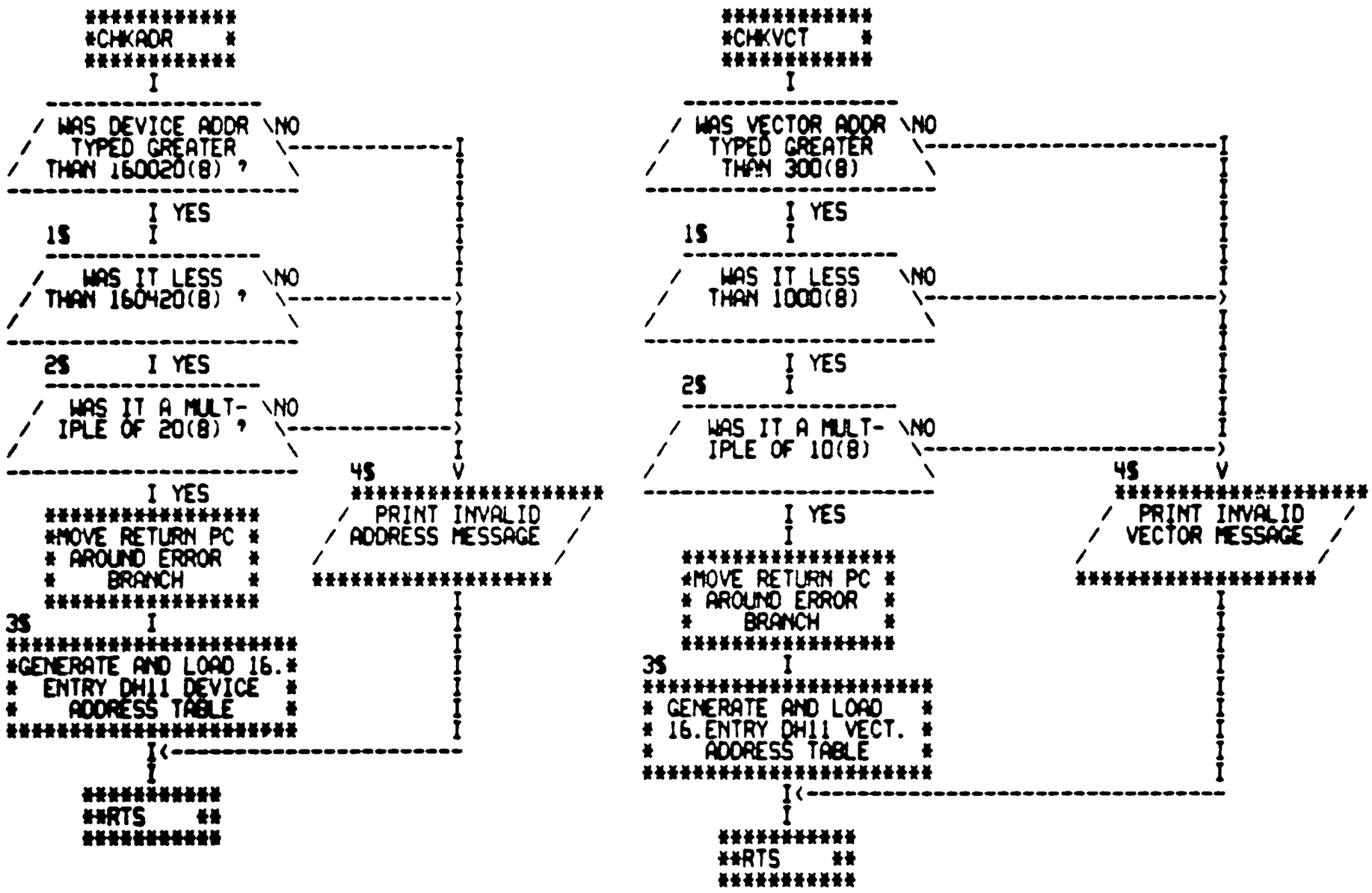
NOTE: I "CHPS1", "CHPS2", AND "SAPS" ARE CALLED WHENEVER
THE MAINLINE CODE HAS TO CLEAR THE PSW, LOCK OUT INTRS,
AND SAVE THE PSW RESPECTIVELY

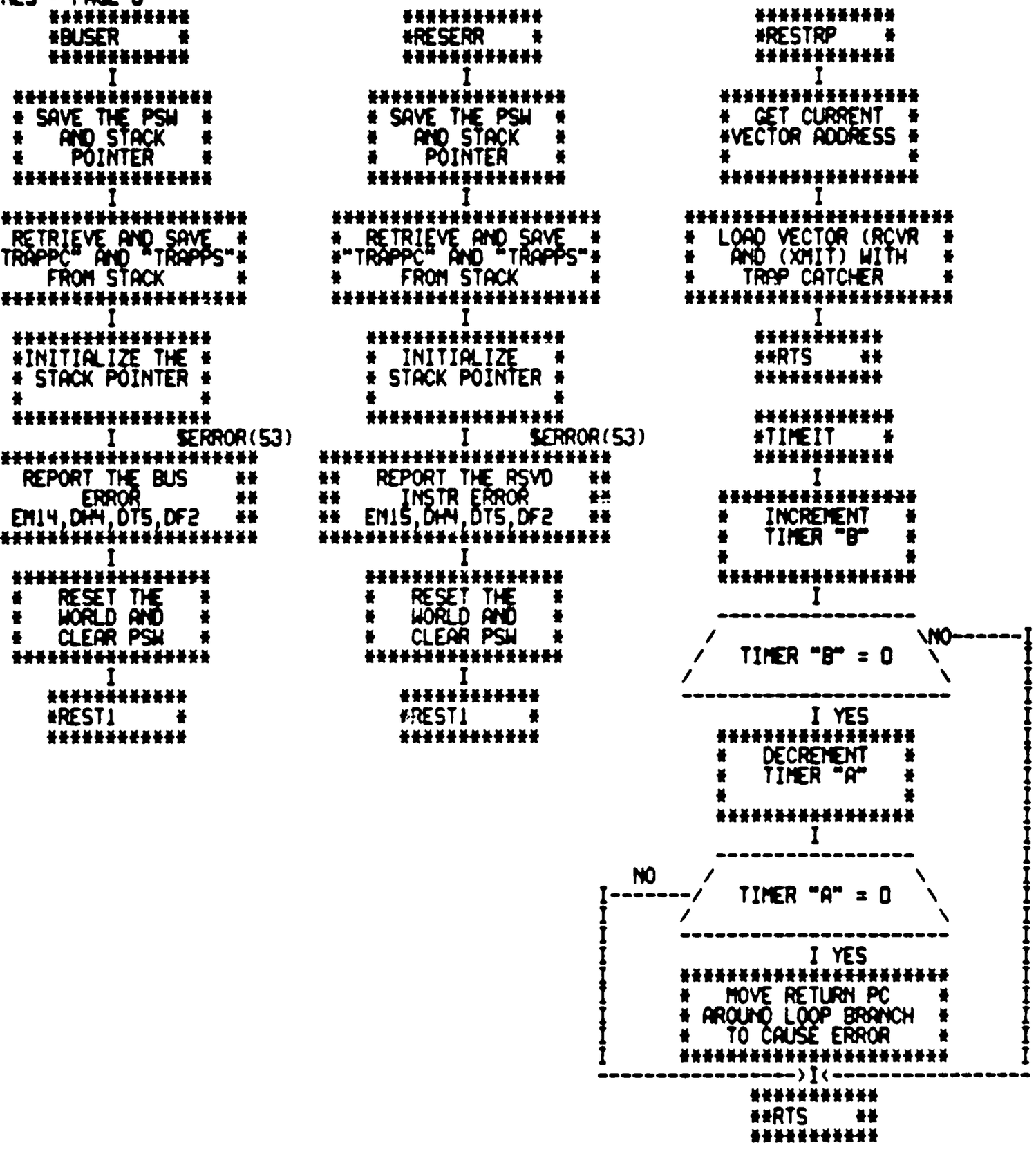
THESE ROUTINES ARE REQUIRED FOR LSI11 COMPATIBILITY

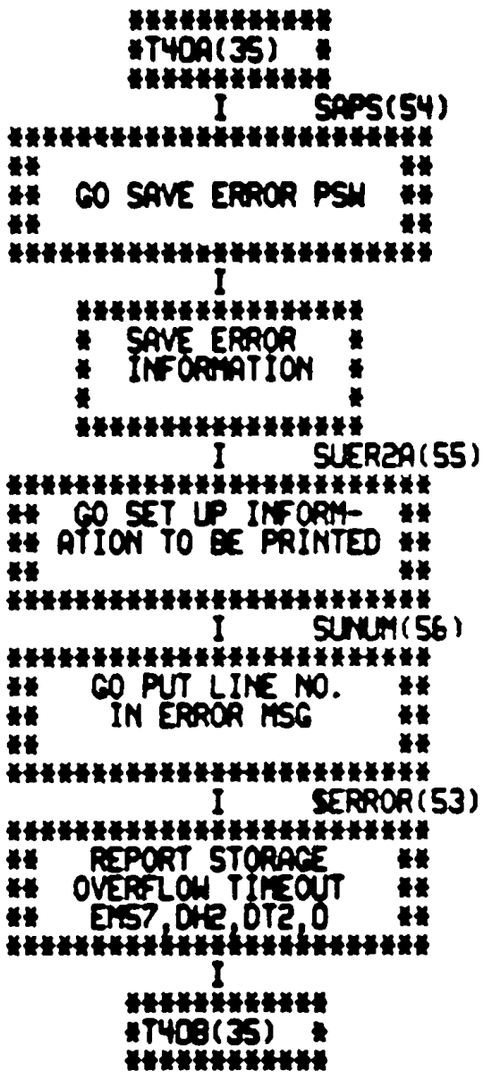

*LDTBF1
SET R1 TO POINT TO XMIT BUFFER
INIT R2=0 AS A CHAR GENERATOR
*LOAD [R2] INTO XMIT BUFFER UPDATE R1
GENERATE NEXT CHAR
LOADED ALL 256.. COMBINATIONS
YES
**RTS
*SUER1
SAPS(54)
GO SAVE PSW IN $TMPO
SAVE TEST NO. IN R0
SAVE R0,R1,R2,R6 IN $REG (0,1,2,6)
CORRECT $REG6
**RTS
*SUER2
SAPS(54)
GO SAVE PSW IN $TMPO
SAVE TEST NO. IN R0
*SAVE R0,R1,R2,R3,R4, AND R6 IN $REG (0,1,2,3,4,6)
CORRECT $REG6
**RTS
*SUER2A
*SUER3
SAVE R0,R1,R2 IN $REG(0,1,2)
**RTS
*SUER4
SAVE LINE NO. IN $TMPO
SAVE R0,R1,R2,R3,R4 IN $REG(0,1,2,3,4)
**RTS

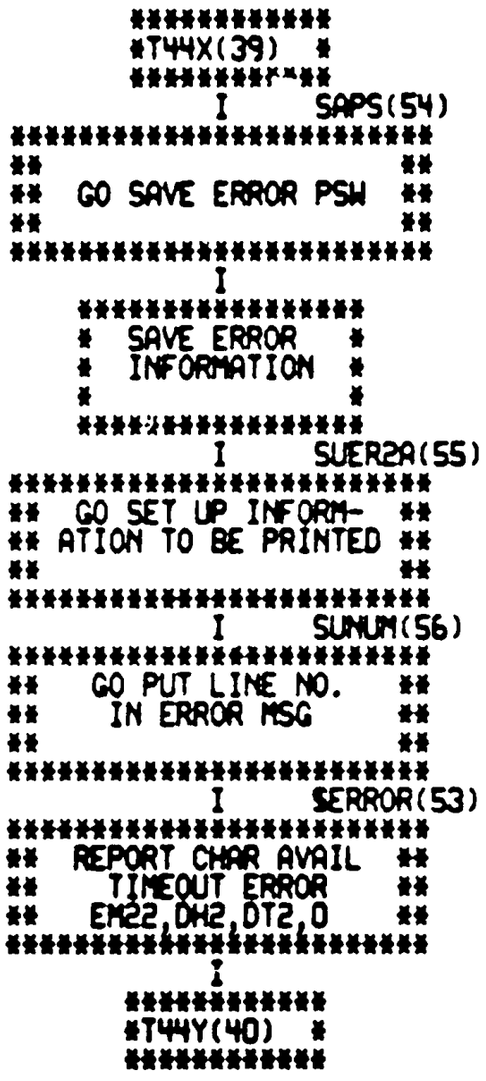

SUNUM
SAVE REGS,RETRIEVE ADDRESSES OF NUM AND MSG BUFFER
GET NUMBER TO CONVERT
CONVERT AND STORE MSD
CONVERT AND STORE LSD
RESTORE REGS
RTS
CLCABC
INIT A LINE COUNTER TO START AT 00
SELECT A LINE IN THE "SCR"
CLEAR "CAR" AND "BCR" FOR SEL LINE
UPDATE LINE COUNTER
CLEARED ALL 16. LINES YET
NO
YES
INIT "SCR" TO SELECT LINE 00
RTS
LDBCR
INIT A LINE COUNTER TO STRT AT 00
SELECT A LINE IN THE "SCR"
LOAD THAT LINE'S "BCR" WITH A 177777
GENERATE NEXT LINE NO.
DONE 16. LINES ?
NO
YES
RTS

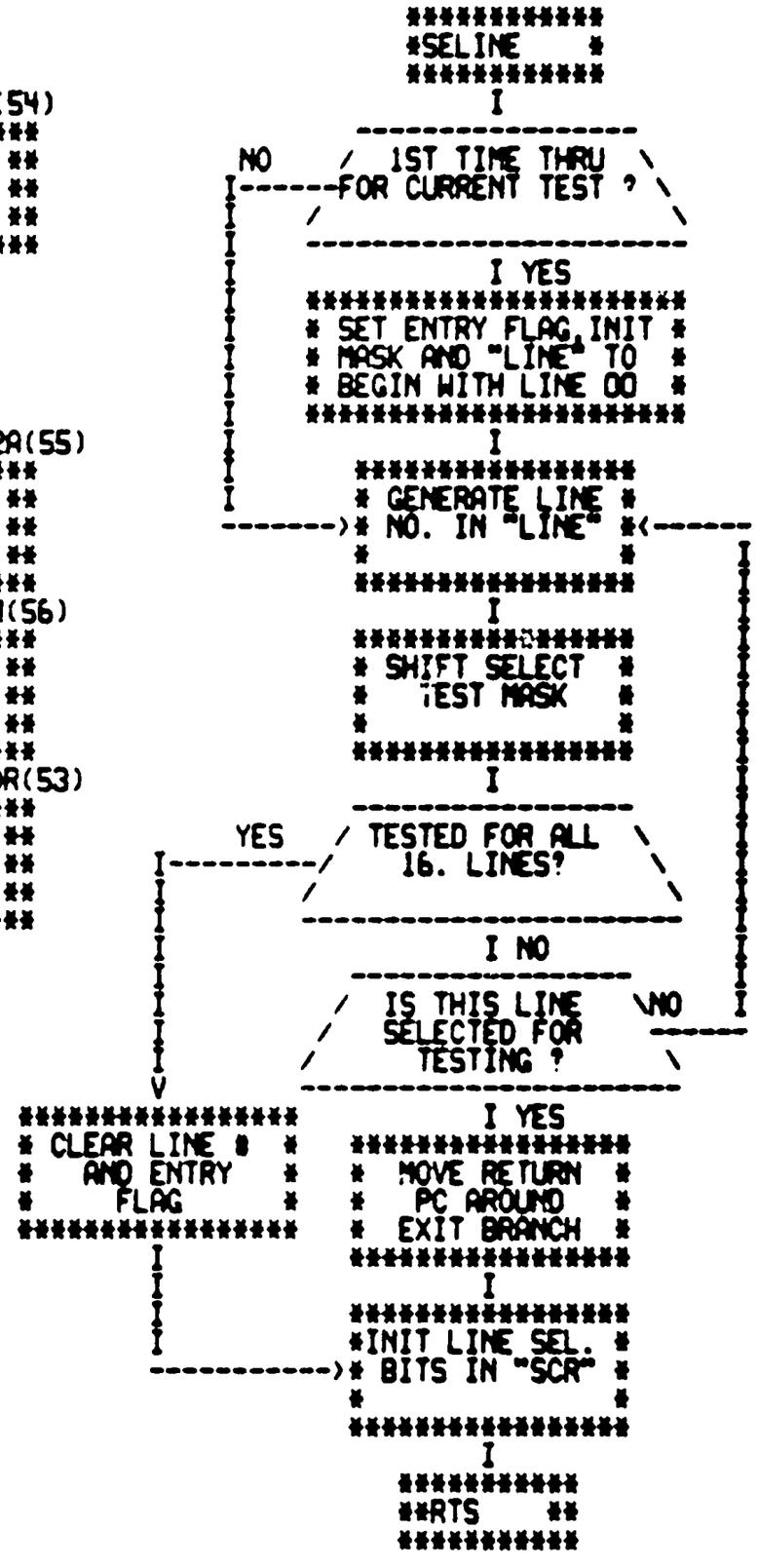

*SUPPAR
INIT LINE COUNTER AND "SCR" TO START AT LINE 00
INIT R3 AND R4 TO LOAD TEST DATA TABLE
LOAD "CAR","BCR", AND"LPR" FOR SELECTED LINE
GENERATE NEW LINE NO.
DONE SET UP FOR 16. LINES
NO
YES
SET UP MULTI-LINE PARITY TEST DATA TABLE
**RTS
*INPARA
ASK USER TO TYPE IN VECTOR DISPLACEMENT
DID HE TYPE A <CR> ONLY ?
YES
NO
DID HE TYPE A "4" ?
DID HE TYPE A "10(8)" ?
LOAD R0 WITH A 20(8)
LOAD R0 WITH A 10(8)
CALL "$EOP" TO REPORT END OF PASS
*ENDA(51)
$SCOPE(52)
"NOP" SCOPE CALL IN $EOP ROUTINE
GENERATE NEXT DEVICE NO.
UPDATE DEVICE PARA-METER TABLE POINTERS
SHIFT SELECT MASK
TESTED ALL POSSIBLE DH11'S
IS THIS DH11 SELECTED FOR TEST
CLEAR $TSTNM TO START AT TST 1 AGAIN
RSTRTA(02)
A
B


*CHKADR *
WAS DEVICE ADDR TYPED GREATER THAN 160020(8) ?
NO
1$
YES
WAS IT LESS THAN 160420(8) ?
NO
2$
YES
WAS IT A MULT- IPLE OF 20(8) ?
NO
YES
4$
PRINT INVALID ADDRESS MESSAGE
*MOVE RETURN PC * * AROUND ERROR * * BRANCH *
3$
GENERATE AND LOAD 16. * ENTRY DH11 DEVICE * * ADDRESS TABLE *
**RTS **
*CHKVCT *
WAS VECTOR ADDR TYPED GREATER THAN 300(8)
NO
1$
YES
WAS IT LESS THAN 1000(8)
NO
2$
YES
WAS IT A MULT- IPLE OF 10(8)
NO
YES
4$
PRINT INVALID VECTOR MESSAGE
*MOVE RETURN PC * * AROUND ERROR * * BRANCH *
3$
* GENERATE AND LOAD * * 16.ENTRY DH11 VECT. * * ADDRESS TABLE *
**RTS **


*BUSER *
SAVE THE PSW AND STACK POINTER
RETRIEVE AND SAVE "TRAPPC" AND "TRAPPS" FROM STACK
INITIALIZE THE STACK POINTER
$ERROR(53)
REPORT THE BUS ERROR
EM14,DH4,DT5,DF2
RESET THE WORLD AND CLEAR PSW
*REST1 *
*RESERR *
SAVE THE PSW AND STACK POINTER
RETRIEVE AND SAVE "TRAPPC" AND "TRAPPS" FROM STACK
INITIALIZE STACK POINTER
$ERROR(53)
REPORT THE RSVD INSTR ERROR
EM15,DH4,DT5,DF2
RESET THE WORLD AND CLEAR PSW
*REST1 *
*RESTRP *
GET CURRENT VECTOR ADDRESS
LOAD VECTOR (RCVR AND (XMIT) WITH TRAP CATCHER
**RTS **
*TIMEIT *
INCREMENT TIMER "B"
TIMER "B" = 0
NO
I YES
DECREMENT TIMER "A"
TIMER "A" = 0
NO
I YES
MOVE RETURN PC AROUND LOOP BRANCH TO CAUSE ERROR
**RTS **


*T40A(35) *
SAPS(54)
** **
** GO SAVE ERROR PSW **
** **
SAVE ERROR INFORMATION
SUER2A(55)
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
** **
SUNUM(56)
** GO PUT LINE NO. **
** IN ERROR MSG **
** **
SERROR(53)
** REPORT STORAGE **
** OVERFLOW TIMEOUT **
** EM57,DH2,DT2,0 **
*T40B(35) *


*T44X(39) *
SAPS(54)
** **
** GO SAVE ERROR PSW **
** **
SAVE ERROR INFORMATION
SUER2A(55)
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
** **
SUNUM(56)
** GO PUT LINE NO. **
** IN ERROR MSG **
** **
SERROR(53)
** REPORT CHAR AVAIL **
** TIMEOUT ERROR **
** EM22,DH2,DT2,0 **
*T44Y(40) *


*SELINE *
1ST TIME THRU FOR CURRENT TEST ?
NO
YES
SET ENTRY FLAG,INIT MASK AND "LINE" TO BEGIN WITH LINE 00
GENERATE LINE NO. IN "LINE"
SHIFT SELECT TEST MASK
TESTED FOR ALL 16. LINES?
NO
IS THIS LINE SELECTED FOR TESTING ?
NO
YES
CLEAR LINE # AND ENTRY FLAG
MOVE RETURN PC AROUND EXIT BRANCH
*INIT LINE SEL. * BITS IN "SCR"
**RTS **

AE1	45	46																
AE2	45	46																
AE3	45	46	46															
AE4	45	46	46															
AE5	45	45	46															
BEGINA	01	01																
BRK1	48	49																
BRK2	48	49																
BRK3	48	49																
BRK4	48	49	49															
BRK5	48	49																
BUSER	59																	
CHKADR	02	58																
CHKVCT	02	58																
CHPS1	54																	
CHPS2	54																	
CLCABC	28	30	50	56														
ENDA	51	57																
HALT	02	53																
INPAR	01	02																
INPAR3	01	02	02															
INPARA	01	57																
LDBCR	30	56																
LDTBF1	02	55																
NPR1A	28	29																
NPR2A	28	29																
NPR3A	28	29																
NPR4A	28	29																
NPR5A	28	29																
NPR6A	28	29																
RESERR	59																	
REST1	59	59																
RESTRP	59																	
RESTRT	01																	
RSTRTA	01	02	57															
RTI	43																	
RTS	54	54	54	54	55	55	55	55	55	56	56	56	57	57	58	58	59	59
	60																	
SAPS	54	55	55	60	60													
SELINE	14	15	16	17	18	19	20	21	22	28	30	35	36	37	38	39	41	44
	45	48	50	51	60													
SETALL	54																	
SLD1	39	40																
SLD2	39	40	40															
SLD3	39	40	40	40														
START	01*																	
START1	01																	
START2	01	01	01	02														
SUER1	03	55																
SUER2	04	05	05	06	07	07	07	08	08	09	09	11	11	12	12	13	13	14
	15	16	16	17	17	22	32	33	38	39	55							
SUER2A	23																	

FLOW CHART CROSS REFERENCE LIST

	23	24	24	25	25	26	26	27	27	28	29	29	29	29	30	31	31	31
	34	34	35	35	36	36	37	37	38	40	40	40	41	41	43	44	44	45
	46	47	48	49	49	49	50	50	51	51	55	60	60					
SUER3	55																	
SUER4	18	19	20	21	30	30	55											
SUNUM	02	14	15	18	19	20	21	22	28	29	30	35	35	36	36	37	37	38
	38	39	40	40	40	41	41	43	43	44	44	45	45	46	47	48	49	49
	49	50	50	51	51	56	60	60										
SUPPAR	42	57																
T40A	35	60																
T40B	35	60																
T44X	38	60																
T44Y	40	60																
T46A	42	43																
T46B	42	43	43	43														
TIMEIT	28	30	35	36	37	38	39	41	42	44	45	47	48	48	50	51	59	
TN	52	52																
TST1	02	03																
TST10	10																	
TST11	11																	
TST12	12																	
TST13	13																	
TST14	14																	
TST15	15																	
TST16	16																	
TST17	17																	
TST2	04																	
TST20	18																	
TST21	19																	
TST22	20																	
TST23	21																	
TST24	22																	
TST25	23																	
TST26	24																	
TST27	25																	
TST3	05																	
TST30	26																	
TST31	27																	
TST32	28																	
TST33	30																	
TST34	31																	
TST35	32																	
TST36	33																	
TST37	34																	
TST4	06																	
TST40	35																	
TST41	36																	
TST42	37																	
TST43	38																	
TST44	38																	
TST45	41																	
TST46	42																	
TST47	42																	

```
                                .NLIST  CND,MD,MC
                                .LIST   TOC,ME,SEQ,BIN
                        165400  $SWR=165400
                                .MCALL  .HEADER,.SCMTAG,.SETUP,.SERROR
                                .MCALL  .SWRHI,.SWRLO,.$CATCH,.EQUATE,SETUP,.SEOP
                                .MCALL  .$SCOPE,.SERRTYP,.STYPOCT,.STYPDEC,.STYPE,.SPOWER,.STRAP
                                .MCALL  .$READ,.$RDOCT
                                .MCALL  .SACT11,.SAPTHDR,.SAPTBLS,.SAPTYPE
                                .ENABLE ABS
                                .TITLE  MAINDEC-11-DZDHM-A
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY ED CROWLEY
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
                                ;*
                        000001  $TN=1

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>

                                .SBTTL  TRAP CATCHER

                        000000          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                        000174          .=174
000174  000000                  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  023474                  JMP     @#INPARX                ;;JUMP TO STARTING ADDRESS OF PROGRAM
000204  000137  002144                  JMP     @#BEGIN         ;BEGIN EXECUTION WITH DEFAULT PARAMETERS
000210  000137  023464                  JMP     @#INPAPC        ;INPUT PARAMETERS - DEVICE SELECTION ONLY

                                ;*****************************************************************

                                .SBTTL  ACT11 HOOKS
                                ;HOOKS REQUIRED BY ACT11
                        000214          $SVPC=.                 ;SAVE PC
                        000046          .=46
```

```
   (1)  000046  020046                    $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
   (1)          000052                    .=52
   (1)  000052  120000                    .WORD   120000          ;;2)SET LOC.52 TO 120000
   (1)          000214                    .=$SVPC                 ;; RESTORE PC
  2252                             ;*****************************************************************
   (1)
   (1)                             .SBTTL  APT PARAMETER BLOCK
   (1)                             ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
   (2)                             ;*****************************************************************
   (1)          000214                    .$X=.   ;;SAVE CURRENT LOCATION
   (1)          000024                    .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
   (1)  000024  000200                    200     ;;FOR APT START UP
   (1)          000044                    .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
   (1)  000044  000214                    $APTHDR ;;POINT TO APT HEADER BLOCK
   (1)          000214                    .=.$X   ;;RESET LOCATION COUNTER
   (2)                             ;*****************************************************************
   (1)                             ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
   (1)                             ;INTERFACE SPEC.
   (1)
   (1)  000214                     $APTHD:
   (1)  000214  000000             $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
   (1)  000216  001230             $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
   (1)  000220  000036             $TSTM:  .WORD   30.     ;;RUN TIM OF LONGEST TEST
   (1)  000222  000170             $PASTM: .WORD   120.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
   (1)  000224  000170             $UNITM: .WORD   120.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
   (1)  000226  000052                     .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
  2257
  2258
   (1)                             .SBTTL  BASIC DEFINITIONS
   (1)
   (1)                             ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
   (1)          001100             STACK=  1100
   (1)                             .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
   (1)                             .EQUIV  TOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
   (1)          177776             PS=     177776          ;;PROCESSOR STATUS WORD
   (1)                             .EQUIV  PS,PSW
   (1)          177774             STKLMT= 177774          ;;STACK LIMIT REGISTER
   (1)          177772             PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
   (1)          177570             DSWR=   177570          ;;HARDWARE SWITCH REGISTER
   (1)          177570             DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
   (1)
   (1)                             ;*GENERAL PURPOSE REGISTER DEFINITIONS
   (1)          000000             R0=     %0              ;;GENERAL REGISTER
   (1)          000001             R1=     %1              ;;GENERAL REGISTER
   (1)          000002             R2=     %2              ;;GENERAL REGISTER
   (1)          000003             R3=     %3              ;;GENERAL REGISTER
   (1)          000004             R4=     %4              ;;GENERAL REGISTER
   (1)          000005             R5=     %5              ;;GENERAL REGISTER
   (1)          000006             R6=     %6              ;;GENERAL REGISTER
   (1)          000007             R7=     %7              ;;GENERAL REGISTER
   (1)                             .EQUIV  R6,SP           ;;STACK POINTER
   (1)                             .EQUIV  R7,PC           ;;PROGRAM COUNTER
   (1)
   (1)                             ;*PRIORITY LEVEL DEFINITIONS
```

```
(1)  000000  PR0=   0          ;;PRIORITY LEVEL 0
(1)  000040  PR1=   40         ;;PRIORITY LEVEL 1
(1)  000100  PR2=   100        ;;PRIORITY LEVEL 2
(1)  000140  PR3=   140        ;;PRIORITY LEVEL 3
(1)  000200  PR4=   200        ;;PRIORITY LEVEL 4
(1)  000240  PR5=   240        ;;PRIORITY LEVEL 5
(1)  000300  PR6=   300        ;;PRIORITY LEVEL 6
(1)  000340  PR7=   340        ;;PRIORITY LEVEL 7
(1)
(1)                            .;*"SWITCH REGISTER" SWITCH DEFINITIONS
(1)  100000  SW15=  100000
(1)  040000  SW14=  40000
(1)  020000  SW13=  20000
(1)  010000  SW12=  10000
(1)  004000  SW11=  4000
(1)  002000  SW10=  2000
(1)  001000  SW09=  1000
(1)  000400  SW08=  400
(1)  000200  SW07=  200
(1)  000100  SW06=  100
(1)  000040  SW05=  40
(1)  000020  SW04=  20
(1)  000010  SW03=  10
(1)  000004  SW02=  4
(1)  000002  SW01=  2
(1)  000001  SW00=  1
(1)                            .EQUIV  SW09,SW9
(1)                            .EQUIV  SW08,SW8
(1)                            .EQUIV  SW07,SW7
(1)                            .EQUIV  SW06,SW6
(1)                            .EQUIV  SW05,SW5
(1)                            .EQUIV  SW04,SW4
(1)                            .EQUIV  SW03,SW3
(1)                            .EQUIV  SW02,SW2
(1)                            .EQUIV  SW01,SW1
(1)                            .EQUIV  SW00,SW0
(1)
(1)                            .;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)  100000  BIT15=  100000
(1)  040000  BIT14=  40000
(1)  020000  BIT13=  20000
(1)  010000  BIT12=  10000
(1)  004000  BIT11=  4000
(1)  002000  BIT10=  2000
(1)  001000  BIT09=  1000
(1)  000400  BIT08=  400
(1)  000200  BIT07=  200
(1)  000100  BIT06=  100
(1)  000040  BIT05=  40
(1)  000020  BIT04=  20
(1)  000010  BIT03=  10
(1)  000004  BIT02=  4
(1)  000002  BIT01=  2
(1)  000001  BIT00=  1
```

```
(1)                             .EQUIV  BIT09,BIT9
(1)                             .EQUIV  BIT08,BIT8
(1)                             .EQUIV  BIT07,BIT7
(1)                             .EQUIV  BIT06,BIT6
(1)                             .EQUIV  BIT05,BIT5
(1)                             .EQUIV  BIT04,BIT4
(1)                             .EQUIV  BIT03,BIT3
(1)                             .EQUIV  BIT02,BIT2
(1)                             .EQUIV  BIT01,BIT1
(1)                             .EQUIV  BIT00,BIT0
(1)
(1)                             ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)          000004             ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)          000010             RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)          000014             TBITVEC=14              ;;T-BIT
(1)          000014             TRTVEC= 14              ;;TRACE TRAP
(1)          000014             BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)          000020             IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)          000024             PWRVEC= 24              ;;POWER FAIL
(1)          000030             EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)          000034             TRAPVEC=34              ;;"TRAP" TRAP
(1)          000060             TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)          000064             TPVEC=  64              ;;TTY PRINTER VECTOR
(1)          000240             PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
2259
 (1)
 (1)                                ;*****************************************************************
 (1)
 (1)                                .SBTTL  COMMON TAGS
 (1)
 (1)                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
 (1)                                ;*USED IN THE PROGRAM.
 (1)
 (1)          001100                        .=1100
 (1)  001100                        $CMTAG:                         ;;START OF COMMON TAGS
 (1)  001100  000000                        .WORD   0
 (1)  001102      000               $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
 (1)  001103      000               $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
 (1)  001104  000000                $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
 (1)  001106  000000                $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
 (1)  001110  000000                $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
 (1)  001112  000000                $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
 (1)  001114      000               $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
 (1)  001115      001               $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
 (1)  001116  000000                $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 (1)  001120  000000                $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
 (1)  001122  000000                $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
 (1)  001124  000000                $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
 (1)  001126  000000                $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
 (1)  001130  000000                        .WORD   0               ;;RESERVED--NOT TO BE USED
 (1)  001132  000000                        .WORD   0
 (1)  001134  000000                        .WORD   0
 (1)  001136  177570                SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
 (1)  001140  177570                DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
 (1)  001142  177560                $TKS:   177560                  ;;TTY KBD STATUS
 (1)  001144  177562                $TKB:   177562                  ;;TTY KBD BUFFER
 (1)  001146  177564                $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
 (1)  001150  177566                $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
 (1)  001152      000               $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
 (1)  001153      002               $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 (1)  001154      012               $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 (1)  001155      000               $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 (1)  001156  000000                $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
 (1)                                                                ;;WHICH  ($REGO) WAS OBTAINED
 (3)  001160  000000                $REGO:  .WORD   0               ;;CONTAINS (($REGAD)+0)
 (3)  001162  000000                $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
 (3)  001164  000000                $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
 (3)  001166  000000                $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
 (3)  001170  000000                $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
 (3)  001172  000000                $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
 (3)  001174  000000                $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
 (3)  001176  000000                $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
 (3)  001200  000000                $TMP0:  .WORD   0               ;;USER DEFINED
 (3)  001202  000000                $TMP1:  .WORD   0               ;;USER DEFINED
 (3)  001204  000000                $TMP2:  .WORD   0               ;;USER DEFINED
 (3)  001206  000000                $TMP3:  .WORD   0               ;;USER DEFINED
 (3)  001210  000000                $TMP4:  .WORD   0               ;;USER DEFINED
 (3)  001212  000000                $TMP5:  .WORD   0               ;;USER DEFINED
 (3)  001214  000000                $TMP6:  .WORD   0               ;;USER DEFINED
 (3)  001216  000000                $TMP7:  .WORD   0               ;;USER DEFINED
```

```
(1)  001220  000000                $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
(1)  001222  000000                $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
(1)  001224     077                $QUES:  .ASCII  /?/             ;;QUESTION MARK
(1)  001225     015                $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
(1)  001226  000012                $LF:    .ASCIZ  <12>            ;;LINE FEED
```

```
(2)                                     ;*****************************************************************
(3)                                     ;*****************************************************************
(2)
(2)                                     .SBTTL  APT MAILBOX-ETABLE
(2)
(2)
(2)                                             .EVEN
(2)  001230                             $MAIL:                  ;;APT MAILBOX
(2)  001230  000000                     $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
(2)  001232  000000                     $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
(2)  001234  000000                     $TESTN: .WORD   ATESTN  ;;TEST NUMBER
(2)  001236  000000                     $PASS:  .WORD   APASS   ;;PASS COUNT
(2)  001240  000000                     $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
(2)  001242  000000                     $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
(2)  001244  000000                     $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
(2)  001246  000000                     $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
(2)  001250                             $ETABLE:                ;;APT ENVIRONMENT TABLE
(2)  001250  000                        $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
(2)  001251  000                        $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
(2)  001252  000000                     $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
(2)  001254  000000                     $USWR:  .WORD   AUSWR   ;;USER SWITCHES
(2)  001256  000000                     $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
(2)                                     ;*                      BITS 15-11=CPU TYPE
(2)                                     ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)                                     ;*                              11/70=06,PDQ=07,Q=10
(2)                                     ;*                      BIT 10=REAL TIME CLOCK
(2)                                     ;*                      BIT  9=FLOATING POINT PROCESSOR
(2)                                     ;*                      BIT  8=MEMORY MANAGEMENT
(2)  001260  000                        $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001261  000                        $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
(2)                                     ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
(2)                                     ;*                              900 NSEC CORE=001
(2)                                     ;*                              300 NSEC BIPOLAR=002
(2)                                     ;*                              500 NSEC MOS=003
(2)  001262  000000                     $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                                     ;*                      MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2)  001264  000                        $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001265  000                        $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001266  000000                     $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001270  000                        $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001271  000                        $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001272  000000                     $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001274  000                        $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001275  000                        $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001276  000000                     $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  001300  000                        $VECT1: .BYTE   AVECT1  ;;INTERRUPT VECTOR#1
(2)  001301  000                        $VECT2: .BYTE   AVECT2  ;;INTERRUPT VECTOR#2
(2)  001302  000                        $PRIOR: .BYTE   APRIOR  ;;BUS PRIORITY #1,#2
(2)  001303  000                                .BYTE   0       ;;SPARE,NOT USED
(2)                                             .EVEN
(2)  001304  000000                     $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001306  000000                     $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
(2)  001310  000000                     $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
```

```
(2)  001312  000000                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
(2)  001314  000000                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
(2)  001316  000000                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
(2)  001320  000000                $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
(2)  001322  000000                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
(2)  001324  000000                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
(2)  001326  000000                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
(2)  001330  000000                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
(2)  001332  000000                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
(2)  001334  000000                $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
(2)  001336  000000                $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
(2)  001340  000000                $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
(2)  001342  000000                $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
(2)  001344  000000                $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
(2)  001346  000000                $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
(2)  001350  000000                $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
(2)  001352  000000                $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
(2)
(2)
(2)  001354                        $ETEND:

(1)
(1)                                .SBTTL  ERROR POINTER TABLE
(1)
(1)                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1)                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1)                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1)                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(1)                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1)                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
(1)                                ;*      DH              ;;POINTS TO THE DATA HEADER
(1)                                ;*      DT              ;;POINTS TO THE DATA
(1)                                ;*      DF              ;;POINTS TO THE DATA FORMAT
(1)
(1)
(1)  001354                        $ERRTB:
2260                               ;ERROR TABLE ITEM FOR ERROR MESSAGE 1
2261
2262  001354  025202                       EM1             ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
2263  001356  025251                       DH1             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR "
2264  001360  025330                       DT1             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2
2265  001362  000000                       0               ;PRINT ALL OCTAL
2266
2267                               ;ERROR TABLE ITEM FOR ERROR MESSAGE 2
2268
2269  001364  025346                       EM2             ;"SYSTEM CONTROL REGISTER ERROR"
2270  001366  025404                       DH2             ; (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2271  001370  025502                       DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2272  001372  000000                       0               ;PRINT ALL OCTAL
2273
2274                               ;ERROR TABLE ITEM FOR ERROR MESSAGE 3
2275
2276  001374  025524                       EM3             ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"
```

```
2277  001376  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2278  001400  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2279  001402  000000                    0               ;PRINT ALL OCTAL
2280
2281                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 4
2282
2283  001404  025602                    EM4             ;"LINE PARAMETER REGISTER ERROR"
2284  001406  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2285  001410  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2286  001412  000000                    0               ;PRINT ALL OCTAL
2287
2288                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 5
2289
2290  001414  025640                    EM5             ;"BREAK CONTROL REGISTER ERROR"
2291  001416  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2292  001420  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2293  001422  000000                    0               ;PRINT ALL OCTAL
2294
2295                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 6
2296
2297  001424  025675                    EM6             ;"SILO STATUS REGISTER ERROR"
2298  001426  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2299  001430  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2300  001432  000000                    0               ;PRINT ALL OCTAL
2301
2302                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 7
2303
2304  001434  025730                    EM7             ;"CURRENT ADDRESS REGISTER ERROR - LINE #XX"
2305  001436  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2306  001440  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2307  001442  000000                    0               ;PRINT ALL OCTAL
2308
2309                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 10
2310
2311  001444  026002                    EM10            ;"BYTE COUNTER REGISTER ERROR - LINE #XX"
2312  001446  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2313  001450  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2314  001452  000000                    0               ;PRINT ALL OCTAL
2315
2316                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 11
2317
2318  001454  026051                    EM11            ;"UNEXPECTED DH11 RCVR INTERRUPT"
2319  001456  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2320  001460  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2321  001462  000000                    0               ;PRINT ALL OCTAL
2322
2323                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 12
2324
2325  001464  026110                    EM12            ;"UNEXPECTED DH11 XMITTR INTERRUPT"
2326  001466  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2327  001470  025502                    DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2328  001472  000000                    0               ;PRINT ALL OCTAL
2329
2330                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 13
```

```
2331
2332  001474  026151                    EM13            ;"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
2333  001476  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2334  001500  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2335  001502  000000                    0               ;PRINT ALL OCTAL
2336
2337                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 14
2338
2339  001504  026232                    EM14            ;"TRANSMITTER NPR LOGIC ERROR - LINE #  "
2340  001506  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2341  001510  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2342  001512  000000                    0               ;PRINT ALL OCTAL
2343
2344                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 15
2345
2346  001514  026301                    EM15            ;"XMITTR FAILED TO INTERRUPT - LINE #  "
2347  001516  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2348  001520  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2349  001522  000000                    0               ;PRINT ALL OCTAL
2350
2351                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 16
2352
2353  001524  026347                    EM16            ;"RCVR FAILED TO INTERRUPT"
2354  001526  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2355  001530  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2356  001532  000000                    0               ;PRINT ALL OCTAL
2357
2358                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 17
2359
2360  001534  026400                    EM17            ;"TRANSMITTER TIMING ERROR - LINE #  "
2361  001536  026444                    DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
2362  001540  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2363  001542  000000                    0               ;PRINT ALL OCTAL
2364
2365                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 20
2366
2367  001544  026542                    EM20            ;RECEIVER TIMING ERROR - LINE #  "
2368  001546  026444                    DH6             ;" (PC)  (PS)  (SP)  TEST  DEVADR  SPEED   TIMEB   TIMEC"
2369  001550  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2370  001552  000000                    0               ;PRINT ALL OCTAL
2371
2372                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 21
2373
2374  001554  026603                    EM21            ;"RCVR FAILED TO INTERRUPT - LINE #  "
2375  001556  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2376  001560  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3 SREG4
2377  001562  000000                    0               ;PRINT ALL OCTAL
2378
2379                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 22
2380
2381  001564  026647                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
2382  001566  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2383  001570  025502                    DT2             ;SERRPC,STMPO,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
2384  001572  000000                    0               ;PRINT ALL OCTAL
```

```
2385
2386                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 23
2387
2388  001574  026723                            EM23            ;"BASIC DATA TEST ERROR - LINE # "
2389  001576  026764                            DH7             ;" (PC)  (PS)  (SP)  TEST  DEVADR  CHRLNG  WAS  S/B "
2390  001600  025502                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2391  001602  000000                            0               ;PRINT ALL OCTAL
2392
2393
2394                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 24
2395
2396  001604  027061                            EM24            ;"AUTO ECHO TEST ERROR - LINE # "
2397  001606  025404                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2398  001610  025502                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2399  001612  000000                            0               ;PRINT ALL OCTAL
2400
2401                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 25
2402
2403  001614  027121                            EM25            ;"BREAK BIT TEST ERROR - LINE # "
2404  001616  025404                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2405  001620  025502                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2406  001622  000000                            0               ;PRINT ALL OCTAL
2407
2408                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 26
2409
2410  001624  027161                            EM26            ;"HALF-DUPLEX TEST ERROR - LINE # "
2411  001626  025404                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2412  001630  025502                            DT2             ;$ERRPC,$TMPO,$REG6,$REG7,$REG1,$REG2,$REG3,$REG4
2413  001632  000000                            0               ;PRINT ALL OCTAL
2414
2415                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 27
2416
2417  001634  027223                            EM27            ;"UNEXPECTED BUS ERROR TRAP"
2418  001636  027255                            DH3             ;  (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
2419  001640  027334                            DT3             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
2420  001642  000000                            0               ;PRINT ALL OCTAL
2421
2422                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 30
2423
2424  001644  027352                            EM30            ;"UNEXPECTED RSVD INSTR TRAP"
2425  001646  027255                            DH3             ;  (PC)  (PS)  (SP)  TEST  TRPPC  TRPPS
2426  001650  027334                            DT3             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2
2427  001652  000000                            0               ;PRINT ALL OCTAL
2428
2429                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 31
2430
2431  001654  027405                            EM31            ;"AUTO ECHO DATA COMPARE ERROR - LINE # "
2432  001656  027455                            DH4             ;  (PC)   (PS)   (SP)    TEST   WASADR  SBADR   WAS    S/B "
2433  001660  025502                            DT2             ;$ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2434  001662  000000                            0               ;PRINT ALL OCTAL
2435
2436                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 32
2437
2438  001664  027552                            EM32            ;"AUTO ECHO TEST TIMEOUT - LINE # "
```

```
2439  001666  027614                    DH5             ;" (PC)  (LPRG)   TEST"
2440  001670  027642                    DT4             ;$ERRPC,$TMP0,$TMP2
2441  001672  000000                    0               ;PRINT ALL OCTAL
2442
2443
2444                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 33
2445
2446  001674  027652                    EM33            ;"PARITY LOGIC TEST ERROR - LINE #  "
2447  001676  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B"
2448  001700  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2449  001702  000000                    0               ;PRINT ALL OCTAL
2450
2451                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 34
2452
2453  001704  027715                    EM34            ;"MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  "
2454  001706  027456                    DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
2455  001710  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2456  001712  000000                    0               ;PRINT ALL OCTAL
2457
2458                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 35
2459
2460  001714  030010                    EM35            ;"MULTI-LINE PARITY DATA TEST TIMEOUT"
2461  001716  030054                    DH14            ;" (PC)  (LPRG)  LINACT "
2462  001720  030104                    DT6             ;"$ERRPC,$TMP0,$TMP3"
2463  001722  000000                    0               ;PRINT ALL OCTAL
2464
2465                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 36
2466
2467  001724  030114                    EM36            ;CHAR AVAILABLE TIMEOUT"
2468  001726  027614                    DH5             ;" (PC)  (LPRG)   TEST"
2469  001730  027642                    DT4             ;$ERRPC,$TMP0,$TMP2
2470  001732  000000                    0               ;PRINT ALL OCTAL
2471
2472                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 37
2473
2474  001734  030143                    EM37            ;"DATA COMPARE ERROR - LINE #  "
2475  001736  027455                    DH4             ;" (PC)  (PS)  (SP)  TEST  WASADR  SBADR  WAS  S/B "
2476  001740  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2477  001742  000000                    0               ;PRINT ALL OCTAL
2478
2479                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 40
2480
2481  001744  030201                    EM40            ;"BUFFER ACTIVE REG ERROR - LINE #  "
2482  001746  025404                    DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2483  001750  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2484  001752  000000                    0               ;PRINT ALL OCTAL
2485
2486                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 41
2487
2488  001754  030244                    EM41            ;"RCVR FALSE INTERRUPT"
2489  001756  027614                    DH5             ;" (PC)  (LPRG)   TEST"
2490  001760  027642                    DT4             ;$ERRPC,$TMP0,$TMP2
2491  001762  000000                    0               ;PRINT ALL OCTAL
2492
```

```
2493                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 42
2494
2495  001764  030271                            EM42            ;"SILO OVERFLOW ERROR"
2496  001766  027614                            DH5             ;" (PC)  (LPRG)   TEST"
2497  001770  027642                            DT4             ;$ERRPC,$TMP0,$TMP2
2498  001772  000000                            0               ;PRINT ALL OCTAL
2499
2500                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 43
2501
2502  001774  030315                            EM43            ;"SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
2503  001776  025404                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2504  002000  025502                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2505  002002  000000                            0               ;PRINT ALL OCTAL
2506
2507                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 44
2508
2509  002004  030375                            EM44            ;"NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
2510  002006  025404                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2511  002010  025502                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2512  002012  000000                            0               ;PRINT ALL OCTAL
2513
2514                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 45
2515
2516  002014  030457                            EM45            ;"XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
2517  002016  025404                            DH2             ;" (PC)  (PS)  (SP)  TEST  DEVADR  REGADR  WAS  S/B "
2518  002020  025502                            DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2519  002022  000000                            0               ;PRINT ALL OCTAL
2520
2521                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 46
2522
2523  002024  030535                            EM46            ;"CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  "
2524  002026  030623                            DH10            ;" (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
2525  002030  030720                            DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
2526  002032  000000                            0               ;PRINT ALL OCTAL
2527
2528                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 47
2529
2530  002034  030742                            EM47            ;"BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  "
2531  002036  030623                            DH10            ; (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR   WAS     S/B"
2532  002040  030720                            DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4
2533  002042  000000                            0               ;PRINT ALL OCTAL
2534
2535                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 50
2536
2537  002044  031023                            EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
2538  002046  025404                            DH2             ;" (PC)   (PS)   (SP)   TEST   DEVADR  REGADR   WAS     S/B"
2539  002050  025502                            DT2             ;"$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2540  002052  000000                            0               ;PRINT ALL OCTAL
2541
2542                                    ;ERROR TABLE ITEM FOR ERROR MESSAGE 51
2543
2544  002054  031101                            EM51            ;"NPR LOGIC TEST 2 ERROR"
2545  002056  031130                            DH11            ;" (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B"
2546  002060  030720                            DT5             ;$ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4"
```

```
2547  002062  000000                    0               ;PRINT ALL OCTAL
2548
2549                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 52
2550
2551  002064  031225                    EM52            ;"BASIC DATA COMPARE ERROR"
2552  002066  025404                    DH2             ;" (PC)   (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2553  002070  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2554  002072  000000                    0               ;PRINT ALL OCTAL
2555
2556                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 53
2557
2558  002074  031023                    EM50            ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  "
2559  002076  031256                    DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2560  002100  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2561  002102  000000                    0               ;PRINT ALL OCTAL
2562
2563                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 54
2564
2565  002104  026647                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
2566  002106  031256                    DH12            ;" (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2567  002110  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2568  002112  000000                    0               ;PRINT ALL OCTAL
2569
2570                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 55
2571
2572  002114  026647                    EM22            ;"CHAR AVAIL FAILED TO SET ON TIME - LINE #  "
2573  002116  031353                    DH13            ;" (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B
2574  002120  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4
2575  002122  000000                    0               ;PRINT ALL OCTAL
2576
2577                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 56
2578
2579  002124  031452                    EM56            ;"OVERRUN BIT FAILED TO SET - LINE #  "
2580  002126  025404                    DH2             ;" (PC)   (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2581  002130  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2582  002132  000000                    0               ;PRINT ALL OCTAL
2583
2584                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 57
2585
2586  002134  031517                    EM57            ;"STORAGE OVERFLOW BIT FAILED - LINE #  "
2587  002136  025404                    DH2             ;" (PC)   (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B"
2588  002140  025502                    DT2             ;$ERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4"
2589  002142  000000                    0               ;PRINT ALL OCTAL
2590
```

```
 2593  002144  005000                    BEGIN:  CLR     R0              ;INIT R0 TO INDICATE DEFAULT PARAMETERS
 2594  002146  005067  023016            BEGINA: CLR     TITFLG          ;INIT TITLE MESSAGE FLAG
 2596  002152  012706  001100                    MOV     #$CMTAG,R6      ;FIRST LOCATION TO BE CLEARED
   (1) 002156  005026                            CLR     (R6)+           ;CLEAR MEMORY LOCATION
   (1) 002160  022706  001126                    CMP     #$BOOAT,R6      ;DONE?
   (1) 002164  001374                            BNE     .-6             ;LOOP BACK IF NO
   (1) 002166  012706  001100                    MOV     #STACK,SP       ;SETUP THE STACK POINTER
   (1) 002172  012737  020102  000020            MOV     #$SCOPE,@#IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
   (1) 002200  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;LEVEL 7
   (1) 002206  012737  020370  000030            MOV     #$ERROR,@#EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
   (1) 002214  012737  000340  000032            MOV     #340,@#EMTVEC+2 ;LEVEL 7
   (1) 002222  012737  022402  000034            MOV     #$TRAP,@#TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
   (1) 002230  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
   (1) 002236  012737  022444  000024            MOV     #$PWRDN,@#PWRVEC ;POWER FAILURE VECTOR
   (1) 002244  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;LEVEL 7
   (1) 002252  005067  176742                    CLR     $TIMES          ;INITIALIZE NUMBER OF ITERATIONS
   (1) 002256  005067  176740                    CLR     $ESCAPE         ;CLEAR THE ESCAPE ON ERROR ADDRESS
   (1) 002262  112767  000001  176625            MOVB    #1,$ERMAX       ;ALLOW ONE ERROR PER TEST
   (1) 002270  012767  002270  176610            MOV     #.,$LPADR       ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
   (1) 002276  012767  002276  176604            MOV     #.,$LPERR       ;SETUP THE ERROR LOOP ADDRESS
   (2) 002304  013746  000004                    MOV     @#4,-(SP)       ;SAVE ERROR VECTOR
   (2) 002310  013746  000006                    MOV     @#6,-(SP)
   (2) 002314  012767  002330  175462            MOV     #64$,4          ;SET UP TIME OUT VECTOR
   (2) 002322  005777  176610                    TST     @SWR            ;TRY TO REFERENCE HARDWARE SWR
   (2) 002326  000407                            BR      65$             ;BRANCH IF NO TIMEOUT TRAP OCCURS
   (2) 002330  012767  000176  176600  64$:     MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWR
   (2) 002336  012767  000174  176574            MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG
   (2) 002344  022626                            CMP     (SP)+,(SP)+     ;RESTORE STACK
   (2) 002346  012637  000006            65$:    MOV     (SP)+,@#6       ;RESTORE ERROR VECTOR
   (2) 002352  012637  000004                    MOV     (SP)+,@#4
   (1) 002356  005067  176654                    CLR     $PASS           ;CLEAR PASS COUNT
   (1) 002362  132767  000200  176661            BITB    #APTSIZE,$ENVM  ;TEST USER SIZE UNDER APT
   (1) 002370  001403                            BEQ     3$              ;YES,USE NON-APT SWITCH
   (1) 002372  012767  001252  176536            MOV     #$SWREG,SWR     ;NO,USE APT SWITCH REGISTER
   (1) 002400                            3$:
```

```
2599  002400  012767  023774  175376  START1: MOV     #BUSER,ERRVEC   ;SET UP THE BUS ERROR VECTOR
2600  002406  012767  000340  175372          MOV     #340,ERRVEC+2
2601  002414  012767  024050  175366          MOV     #RESERR,RESVEC  ;SET UP THE RSVD INSTR VECTOR
2602  002422  012767  000340  175362          MOV     #340,RESVEC+2
2603  002430  005767  022534                  TST     TITFLG          ;HAVE WE TYPED TITLE ONCE ?
2604  002434  001004                          BNE     1$              ;BR IF YES
2605  002436  104400                          TYPE                    ;GO TYPE PROGRAM TITLE
2606  002440  031566                          TITLE
2607  002442  005167  022522                  COM     TITFLG          ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
2608  002446  005767  022424          1$:     TST     VCFLG           ;START AT 200 ??
2609  002452  001404                          BEQ     11$             ;BR IF NOT
2610  002454  004767  020740                  JSR     PC,INPARA       ;GO ASK FOR PARAMETERS
2611  002460  005067  022412                  CLR     VCFLG           ;RE INIT VECTOR FLAG
2612  002464  005700                  11$:    TST     R0              ;USE DEFAULT PARAMETERS ?
2613  002466  001407                          BEQ     START2          ;BR IF YES
2614  002470  022700  177777                  CMP     #-1,R0          ;CHANGE DH SELECT PARAM ONLY ?
2615  002474  001002                          BNE     2$              ;BR IF NOT
2616  002476  000167  021044                  JMP     INPAR3          ;GO ASK FOR SELECT PARAM.
2617  002502  000167  021000          2$:     JMP     INPAR           ;GO ASK FOR ALL PARAMETERS
2618
2619  002506  012767  024774  022434  START2: MOV     #DHADTB-2,ADPTR ;GET POINTER TO ADDRESS TABLE
2620  002514  012767  025034  022430          MOV     #DHVCTB-2,VCPTR ;GET POINTER TO VECTOR TABLE
2621  002522  012767  025076  022424          MOV     #BRLVL-2,BRPTR  ;GET POINTER TO BR LEVEL TABLE
2622  002530  012767  177777  022404          MOV     #-1,DHNUM       ;START WITH DH #00
2623  002536  012767  000001  021640          MOV     #1,SELMSK       ;SET UP DH11 BIT TEST MARKER
2624
2625  002544  005267  022372          RESTRT: INC     DHNUM           ;GENERATE DH11 DEV NUMBER
2626  002550  062767  000002  022372          ADD     #2,ADPTR        ;UPDATE TABLE POINTERS
2627  002556  062767  000002  022366          ADD     #2,VCPTR
2628  002564  062767  000002  022362          ADD     #2,BPPTR
2629  002572  036767  021606  021606          BIT     SELMSK,DHSEL    ;TEST FOR SELECTED DH11
2630  002600  001004                          BNE     RSTRTA          ;BR IF SELECTED FOR TEST
2631  002602  006367  021576          REST1:  ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
2632  002606  001737                          BEQ     START2          ;BR IF 16 TESTED - START OVER
2633  002610  000755                          BR      RESTRT          ;GO TEST IF THIS ONE SELECTED
2634  002612  017767  022332  021560  RSTRTA: MOV     @ADPTR,DHADR    ;SET UP DH11 ADDRESS
2635  002620  017767  022326  021554          MOV     @VCPTR,DHVCT    ;SET UP THE DH11 VECTOR ENTRY
2636  002626  017767  022322  022304          MOV     @BRPTR,DHRLVL   ;GET BR LEVEL VALUES
2637  002634  004567  020256                  JSR     R5,SUNUM        ;GO SET DH NUMBER IN THE MESSAGE BUFFER
2638  002640  025142                          DHNUM
2639  002642  031656                          TITLE2+20
2640  002644  104400                          TYPE                    ;GO PRINT "TESTING DH11 #XX"
2641  002646  031636                          TITLE2
2642  002650  004767  017734                  JSR     PC,LDTBF1       ;GO LOAD XMITTR OUTPUT BUFFER WITH
2643                                                                  ;BINARY COUNT PATTERN
2644  002654  012767  002654  176224          MOV     #.,@LPADR       ;INIT SCOPE LOOP RETURN
```

```
 2647                                   ;*****************************************************************
  (3)                                   ;*TEST 1        CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
  (3)                                   ;*****************************************************************
  (2)  002662  000004                   TST1:   SCOPE
  (1)  002664  012767  000001  176342           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 2648                                   .REM    %
  (1)                                   TEST ABSTRACT:
  (1)                                   ********-***********
  (1)
  (1)                                           THIS TEST ATTEMPTS TO REFERENCE EACH OF THE EIGHT REGISTERS
  (1)                                   IN THE DH11 SELECTED FOR TEST USING ITS ASSIGNED UNIBUS ADDRESS. IF
  (1)                                   ANY ADDRESS FAILS TO RESPOND A BUS ERROR TRAP VECTORS THE TEST TO THE
  (1)                                   ERROR SET-UP AND CALL ROUTINE. AFTER THE ERROR IS TYPED THE TEST WILL
  (1)                                   TEST THE NEXT DH11 ADDRESS IN SEQUENCE UNTIL ALL EIGHT ARE TESTED.
  (1)
  (1)                                   ERRORS:
  (1)                                   *******
  (1)
  (1)                                   1.) [ERROR  1] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED
  (1)                                       TO RESPOND WITH "SSYN" WHEN REFERENCED.
  (1)
  (1)                                   SYNC:           (NONE)
  (1)                                   *****
  (1)
  (1)                                   DEBUG:
  (1)                                   ******
  (1)
  (1)                                   1.) PROBLEM IS MOST LIKELY THE M7277 MODULE.
  (1)                                   2.) IF ALL EIGHT REGISTERS FAIL TO RESPOND, MAKE SURE THAT YOU CONFIGURED
  (1)                                       THE PROGRAM PROPERLY BEFORE STARTING.  IF YOU DID, CHECK THE SETTINGS
  (1)                                       OF THE ADDRESS SELECT JUMPERS ON THE M7277 MODULE.
  (1)                                   3.) IF ONE OR MORE RESPONDED PROPERLY, SET UP AN ERROR SCOPE LOOP AND
  (1)                                       BACKTRACK THROUGH THE LOGIC STARTING WITH THE KEY LOGIC SIGNALS
  (1)                                       LISTED BELOW.
  (1)
  (1)                                   KEY LOGIC:
  (1)                                   **********
  (1)
  (1)                                           M7277   SH3     SSYN H                          CE2
  (1)                                                           DEVICE RESPONDING L     E72-6
  (1)                                           SH4     DEVICE SELECTED H       E09-11
  (1)                                   %
 2649  002672  010102                           MOV     R1,R2           ;COPY IT INTO R2
 2650  002674  010205                           MOV     R2,R5           ;ALSO R5
 2651  002676  062705  000016                   ADD     #16,R5          ;R5 WILL TELL US WHEN WE'VE TESTED ALL 8
 2652  002702  016746  175076                   MOV     ERRVEC,-(SP)    ;SAVE BUS ERROR VECTOR
 2653  002706  012767  002734  175070           MOV     #3$,ERRVEC      ;GO TO 3$ IF REG FAILS TO RESPOND
 2654  002714  162702  000002                   SUB     #2,R2           ;SO WE START WITH FIRST REG
 2655
 2656  002720  062702  000002           1$:     ADD     #2,R2           ;POINT TO A DH11 REGISTER
 2657  002724  005712                   2$:     TST     (R2)            ;ACCESS DH11 REG ADDR
 2658  002726  020205                           CMP     R2,R5           ;TESTED ALL EIGHT ??
 2659  002730  001373                           BNE     1$              ;BR IF NOT
 2660  002732  000410                           BR      4$              ;BR WHEN ALL 8 ARE DONE
```

```
2661
2662  002734  004767  017672          3$:   JSR     PC,SUER1        ;GO SET UP ERROR INFO
2663  002740  022626                        CMP     (SP)+,(SP)+     ;FIX SP BECAUSE OF TRAP
2664  002742  012767  002724  176140        MOV     #2$,SLPERR      ;SET UP ERROR LOOP RETURN
2665  002750  104001                        ERROR   1               ;DH11 REGISTER FAILED TO RESPOND TO MSYN
2666
2667  002752  000762                        BR      1$              ;GO TEST NEXT ONE
2668  002754  012667  175024          4$:   MOV     (SP)+,ERRVEC    ;RESTORE BUS ERROR VECTOR
```

```
2671                                    ;*****************************************************************
 (3)                                    ;*TEST 2        TEST THAT "MASTER CLR" CAN CLEAR THE "SCR","LPR","BKR",AND "SSR" REGS
 (3)                                    ;*****************************************************************
 (2)  002760  000004                    TST2:   SCOPE
 (1)  002762  012767  000002  176244            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2672                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST SETS ALL WRITEABLE BITS IN THE TARGET REGISTER
 (1)                                    (SCR, LPR, BKR, AND SSR) THEN SETS BIT11 IN THE "SCR" (MASTER CLEAR)
 (1)                                    AND CHECKS THAT IT INDEED CLEARED ALL BITS IN THE TARGET REGISTER.
 (1)                                    IT PERFORMS THIS SEQUENCE FOR ALL TARGET REGISTERS.
 (1)                                    IF A REGISTER FAILS TO CLEAR PROPERLY, THE ERROR IS REPORTED, AND THEN
 (1)                                    THE ROUTINE TESTS THE NEXT REGISTER IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  3] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED
 (1)                                        TO CLEAR WHEN MASTER CLEAR WAS ACTIVATED.
 (1)
 (1)                                    SYNC:
 (1)                                    *****
 (1)
 (1)                                            M7289   SH6     SCR 11 H (MASTER CLEAR)         FK2
 (1)
 (1)                                    DEBUG:
 (1)                                    **.***
 (1)
 (1)                                    1.) IF THE ERROR REPORTS INDICATE THAT ALL REGISTERS ARE FAILING, ESTABLISH
 (1)                                        AN ERROR SCOPE LOOP AND PROCEED TO BACKTRACK THROUGH THE KEY LOGIC
 (1)                                        SIGNALS LISTED BELOW.
 (1)                                    2.) IF ONLY ONE REGISTER FAILS WITH ALL SET BITS -- THEN LET TESTS 03-10
 (1)                                        RUN -- THEY WILL PROBABLY GIVE BETTER ISOLATION. USE ONE OF THESE
 (1)                                        TESTS TO DEBUG THE FAULT.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7277   SH3     INIT A L                FR2
 (1)                                                            INIT B L                FM2
 (1)                                                            INIT A H                EF2
 (1)                                                            INIT B H                FV2
 (1)                                            SH4     LOAD SCR HIGH BYTE H    CP1
 (1)
 (1)                                            M7289   SH6     SCR 11 H (MASTER CLEAR) FK2
 (1)
 (1)                                            M7278   SH3     BUFF DATA 11 H          AA1
 (1)                                    %
2673  002770  012705  024416                  MOV     #MSTCLR,R5      ;GET POINTER TO ADDRESS TABLE
2674  002774  010125                          MOV     R1,(R5)+        ;SET UP THE TEST ADDRESS TABLE SO THAT
2675  002776  010125                          MOV     R1,(R5)+        ;IT CONTAINS THE ADDRESSES OF THE
2676  003000  010125                          MOV     R1,(R5)+        ;SCR,LPR,BKR, AND SSR REGISTERS
```

```
 2677  003002  010125                        MOV     R1,(R5)+
 2678  003004  062745  000016                ADD     #SSR,-(R5)      ;GENERATE SSR ADDRESS
 2679  003010  062745  000014                ADD     #BKR,-(R5)      ;GENERATE BKR ADDRESS
 2680  003014  062745  000004                ADD     #LPR,-(R5)      ;GENERATE LPR ADDRESS
 2681  003020  005745                        TST     -(R5)           ;POINT R5 TO FIRST ADDR ENTRY (SCR)
 2682  003022  005004                        CLR     R4              ;RESULT S/B 000000 AFTER MASTER CLEAR
 2683
 2684  003024  012502              1$:       MOV     (R5)+,R2        ;GET REG ADDRESS
 2685  003026  022705  024426                CMP     #MSTCLR+10,R5   ;DONE ALL FOUR REGS ??
 2686  003032  001415                        BEQ     TST3            ;;BR IF YES
 2687  003034  012712  177777        2$:     MOV     #-1,(R2)        ;SET 1'S IN REGISTER
 2688  003040  052711  004000                BIS     #BIT11,(R1)     ;ISSUE MASTER CLEAR
 2689  003044  011203                        MOV     (R2),R3         ;GET CONTENT OF REGISTER
 2690  003046  001766                        BEQ     1$              ;;BR IF IT'S ALL ZEROES
 2691
 2692  003050  004767  017616                JSR     PC,SUER2        ;GO SET UP ERROR INFO
 2693  003054  012767  003034  176026        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 2694  003062  104003                        ERROR   3               ;MASTER CLR FAILED TO CLR SEL. REG.
 2695  003064  000757                        BR      1$              ;GO TEST NEXT REGISTER
```

```
2698                                    ;*****************************************************************
 (3)                                    ;*TEST 3        TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)
 (3)                                    ;*****************************************************************
 (2)  003066  000004                    TST3:   SCOPE
 (1)  003070  012767  000003  176136            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2699                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT EACH R/W BIT IN THE "SCR" REGISTER
 (1)                                    CAN BE INDIVIDUALLY SET AND CLEARED IN NORMAL MODE (MAINT BIT = 0)
 (1)                                    A BIT MASK (RGMSK1:  131177) IS USED TO DEFINE THE R/W BITS ( ALL BUT
 (1)                                    BITS 14, 11, 10, 8, AND 7). THE TEST IS REPEATED ELEVEN TIMES WITH A
 (1)                                    DIFFERENT BIT SELECTED FOR EACH TEST. R5 CONTAINS THE BIT CURRENTLY
 (1)                                    BEING TESTED. IF AN ERROR IS DETECTED, IT IS REPORTED AND THEN THE TEST
 (1)                                    RESUMES WITH THE NEXT BIT IN SEQUENCE UNTIL ALL HAVE BEEN TESTED.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  2] IS CALLED TO REPORT A FAILURE TO SET PROPERLY AND
 (1)                                    AGAIN TO REPORT A FAILURE TO CLEAR PROPERLY.
 (1)
 (1)                                    SYNC:
 (1)                                    *****
 (1)
 (1)                                    1.) SET FAILURE         M7277   SH4     LOAD SSR LOW BYTE H     CR1
 (1)
 (1)                                    2.) CLR FAILURE         M7277   SH4     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) IF ALL BITS FAIL - SUSPECT THE "LOAD SCR" SIGNALS ON THE M7277 SH4.
 (1)                                    2.) IF ONLY ONE OR TWO BITS FAIL - SUSPECT EITHER THE "SCR" REGISTER
 (1)                                        FLOPS ON THE M7289 SH6, THE BUS RECEIVERS ON THE M7278 SH3 AND SH4,
 (1)                                        OR THE MULTIPLEXORS AND BUS DRIVERS ON THE M7289 SH5-8.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7278   SH3     BUFF DATA <15:08> H
 (1)                                                    SH4     BUFF DATA <07:00> H
 (1)
 (1)                                            M7277   SH4     LOAD SCR LOW BYTE H     CT2
 (1)                                                            LOAD SCR HIGH BYTE H    CP1
 (1)                                                            DATA TO BUS H           EN2
 (1)                                                            DATA SOURCE <A,B,C> H   DU1,DU2,DT2
 (1)
 (1)                                            M7289   SH5     BUF DATA TO BUS B H     E05-12
 (1)                                                            BUS DATA <15:12> L
 (1)                                                    SH6     BUS DATA <11:08> L
 (1)                                                            SCR <15:00> H
 (1)                                                    SH7     BUF DATA TO BUS A H     E05-8
```

```
(1)                                                    BUS DATA <07:04> L
(1)                                             SH8    BUS DATA <03:00> L
(1)                             ;
2700  003076  012767  003126  176004          MOV     #4$,$LPERR      ;SET UP ERROR LOOP RETURN
2701  003104  010102                          MOV     R1,R2           ;GET REGISTER ADDRESS
2702  003106  012705  000001                  MOV     #1,R5           ;SET UP TO START WITH BIT00
2703
2704  003112  030567  021642          1$:     BIT     R5,RGMSK1       ;SHALL WE TEST THIS BIT ?
2705  003116  001003                          BNE     4$              ;BR IF YES
2706  003120  006305                  2$:     ASL     R5              ;SHIFT TO TST NEXT BIT
2707  003122                          3$:
 (2)  003122  001430                          BEQ     TST4            ;;<BR IF DONE ALL R/W BITS>
2708  003124  000772                          BR      1$              ;GO TEST NEXT BIT
2709
2710  003126  010504                  4$:     MOV     R5,R4           ;RESULT S/B IN R4
2711  003130  005012                          CLR     (R2)            ;INIT REG BEING TESTED
2712  003132  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
2713  003140  010512                          MOV     R5,(R2)         ;SET THE BIT
2714  003142  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2715  003144  020403                          CMP     R4,R3           ;RESULT = S/B DATA ??
2716  003146  001403                          BEQ     5$              ;BR IF YES
2717
2718  003150  004767  017516                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
2719  003154  104002                          ERROR   2               ;SELECTED BIT FAILED TO SET IN SCR
2720
2721  003156  005004                  5$:     CLR     R4              ;SET UP TO CLEAR THE BIT S/B=000000
2722  003160  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2723  003166  040512                          BIC     R5,(R2)         ;CLR THE SELECTED BIT
2724  003170  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2725  003172  001403                          BEQ     6$              ;BR IF IT CLEARED
2726
2727  003174  004767  017472                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
2728  003200  104002                          ERROR   2               ;SELECTED BIT FAILED TO CLEAR IN SCR
2729  003202  000746                  6$:     BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;*****************************************************************
                                ;*TEST 4        TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)
                                ;*****************************************************************
(2)  003204  000004             TST4:   SCOPE
(1)  003206  012767  000004  176020     MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    x
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE "SCR" REGISTER READ ONLY BITS CAN
                                NOT BE SET OR CLEARED IN NORMAL MODE. A BIT MASK (RGMSK2:  046600)
                                IS USED TO DEFINE THE READ ONLY BITSS (14,11,10,8, AND 7). THE TEST
                                IS REPEATED FIVE TIMES, ONCE FOR EACH BIT TO BE TESTED, AND ANY
                                ERRORS DETECTED ARE REPORTED. AFTER THE ERROR REPORT THE TEST RESUMES
                                WITH THE NEXT BIT IN SEQUENCE UNTIL ALL BITS HAVE BEEN TESTED.

                                ERRORS:
                                *******

                                1.) [ERROR  2] IS CALLED TO REPORT ANY READ ONLY BIT THAT FAILED
                                    TO RESPOND PROPERLY.

                                SYNC:
                                *****

                                        M7277   SM4     LOAD SSR LOW BYTE H     CR1

                                DEBUG:
                                ******

                                        SAME AS FOR TEST 03

                                KEY LOGIC:
                                **********

                                        SAME AS FOR TEST 03

                                %
2734  003214  010102                    MOV     R1,R2           ;MAKE IT THE REG. ADDR ALSO
2735  003216  012705  000001            MOV     #1,R5           ;INIT BIT TEST MARKER
2736  003222  030567  021534    1$:     BIT     R5,RGMSK2       ;IS IT A READ ONLY BIT ??
2737  003226  001003                    BNE     3$              ;BR IF IT IS - GO TEST IT
2738  003230  006305            2$:     ASL     R5              ;SHIFT BIT MARKER
2739  003232  001420                    BEQ     TST5            ;;BR IF DONE ALL BITS
2740  003234  000772                    BR      1$              ;GO TEST THIS BIT
2741
2742  003236  005004            3$:     CLR     R4              ;RESULT S/B = 000000
2743  003240  005012                    CLR     (R2)            ;INIT REG BEING TESTED
2744  003242  112761  000000  000016    MOVB    #0,SSR(R1)      ;SCOPE SYNC
2745  003250  010512                    MOV     R5,(R2)         ;ATTEMPT TO SET A READ ONLY BIT
2746  003252  011203                    MOV     (R2),R3         ;GET THE WAS DATA
2747  003254  001765                    BEQ     2$              ;BR IF THE BIT DIDN'T SET
2748
2749  003256  004767  017410            JSR     PC,SUER2        ;GO SET UP ERROR INFO
```

```
 2750  003262  012767  003236  175620          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN ADDR
 2751  003270  104002                          ERROR   2               ;READ ONLY BIT SET IN "SCR"
 2752  003272  000756                          BR      2$              ;CONTINUE WITH NEXT BIT
```

```
2755                                    ;*****************************************************************
 (3)                                    ;*TEST 5        TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
 (3)                                    ;*****************************************************************
 (2)  003274  000004                    TST5:   SCOPE
 (1)  003276  012767  000005  175730            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2756                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ****************
 (1)
 (1)                                            THIS TEST VERIFIES THAT THE "SCR" REGISTER READ-ONLY BITS
 (1)                                    (14, 10, AND 07) CAN BE SET/CLR IN MAINT. MODE (SCR09=1) ONLY.
 (1)                                    A BIT MASK (RGMSK6:  042200) IS USED TO DEFINE THE BITS TO TEST.
 (1)                                    THE TEST PERFORMS THE FOLLOWING SEQUENCE:
 (1)
 (1)                                            1. SELECT A BIT TO TEST
 (1)                                            2. SET MAINT. MODE
 (1)                                            3. SET THE BIT AND VERIFY THAT IT SET
 (1)                                            4. CLEAR THE MAINT. MODE BIT
 (1)                                            5. ATTEMPT TO CLEAR THE TEST BIT
 (1)                                            6. TEST TO SEE THAT IT DID NOT CLEAR
 (1)                                            7. SET MAINT. MODE
 (1)                                            8. CLEAR THE TEST BIT AND VERIFY THAT IT CLEARED
 (1)                                            9. REPEAT 1-7 UNTIL ALL BITS HAVE BEEN TESTED
 (1)
 (1)                                    ANY ERRORS DETECTED ARE REPORTED AND THE TEST RESUMES WITH THE NEXT
 (1)                                    BIT IN SEQUENCE UNTIL ALL BITS HAVE BEEN TESTED.
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  2] IS CALLED AT THREE DIFFERENT POINTS TO REPORT ONE OF
 (1)                                        THE THREE POSSIBLE FAILURE MODES DESCRIBED IN 3, 6, AND 7 IN THE
 (1)                                        ABSTRACT.
 (1)
 (1)                                    SYNC:
 (1)                                    ******
 (1)
 (1)                                    1.) STEP 3 FAILURE TO SET WITH MAINT. MODE SET
 (1)
 (1)                                            M7277   SM4     LOAD SSR LOW BYTE H     CR1
 (1)
 (1)                                    2.) STEP 5 FAILURE TO REMAIN SET WITH MAINT MODE NOT SET
 (1)
 (1)                                            M7277   SM4     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    3.) STEP 8 FAILURE TO CLEAR WITH MAINT. MODE SET
 (1)
 (1)                                            M7277   SM4     LOAD LPR H              EP2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) ASSUMING THE PREVIOUS TESTS RAN SUCCESSFULLY THE FAULT IS MOST
 (1)                                        LIKELY THE M7289 MODULE.
```

```
(1)
(1)                                     KEY LOGIC:
(1)                                     **********
(1)
(1)                                             SAME AS TEST 03 WITH THE FOLLOWING ADDITION
(1)
(1)                                             M7289   SH4     74121 ONE-SHOTS E35-6, E23-6
(1)                                     %
2757  003304  012767  003334  175576            MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
2758  003312  010102                            MOV     R1,R2           ;MAKE IT REG ADDR TOO
2759  003314  012705  000001                    MOV     #1,R5           ;INIT BIT TEST MARKER
2760  003320  030567  021446            1$:     BIT     R5,RGMSK6       ;IS IT A READ ONLY BIT ??
2761  003324  001003                            BNE     3$              ;BR IF YES - TEST IT
2762  003326  006305                    2$:     ASL     R5              ;SHIFT THE BIT MARKER
2763  003330  001457                            BEQ     TST6            ;;BR IF DONE ALL SELECTED BITS
2764  003332  000772                            BR      1$              ;GO TEST FOR THIS BIT
2765
2766  003334  010504                    3$:     MOV     R5,R4           ;SET UP S/B DATA
2767  003336  052704  001000                    BIS     #BIT09,R4       ;PUT IN THE MAINT. BIT
2768  003342  005012                            CLR     (R2)            ;INIT REG BEING TESTED
2769  003344  052712  001000                    BIS     #BIT09,(R2)     ;TURN ON MAINT. MODE
2770  003350  112761  000000  000016            MOVB    #0,SSR(R1)      ;SCOPE SYNC
2771  003356  050512                            BIS     R5,(R2)         ;SET THE SELECTED BIT
2772  003360  011203                            MOV     (R2),R3         ;GET THE WAS DATA
2773  003362  020304                            CMP     R3,R4           ;DID SELECTED BIT GET SET ??
2774  003364  001404                            BEQ     4$              ;BR IF IT DID
2775
2776  003366  004767  017300                    JSR     PC,SUER2        ;GO SET UP ERROR INFO
2777  003372  104002                            ERROR   2               ;SELECTED BIT FAILED TO SET IN MAINT MODE
2778  003374  000754                            BR      2$              ;GO TEST NEXT BIT
2779
2780  003376  042712  001000            4$:     BIC     #BIT09,(R2)     ;TURN OFF MAINT. MODE
2781  003402  042704  001000                    BIC     #BIT09,R4       ;CLR MAINT BIT IN S/B DATA
2782  003406  112761  000000  000017            MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2783  003414  040512                            BIC     R5,(R2)         ;ATTEMPT TO CLR SELECTED BIT
2784  003416  011203                            MOV     (R2),R3         ;GET THE WAS DATA
2785  003420  020304                            CMP     R3,R4           ;DID BIT GET CLEARED ??
2786  003422  001404                            BEQ     5$              ;BR IF IT DIDN'T
2787
2788  003424  004767  017242                    JSR     PC,SUER2        ;GO SET UP ERROR INFO
2789  003430  104002                            ERROR   2               ;SELECTED BIT GOT CLEARED WITH MAINT MODE OFF
2790  003432  000735                            BR      2$              ;GO TEST NEXT BIT
2791
2792  003434  012704  001000            5$:     MOV     #BIT09,R4       ;SET UP S/B DATA
2793  003440  050412                            BIS     R4,(R2)         ;SET MAINT. MODE
2794  003442  012761  000000  000004            MOV     #0,LPR(R1)      ;SCOPE SYNC
2795  003450  040512                            BIC     R5,(R2)         ;NOW CLR SELECTED BIT
2796  003452  011203                            MOV     (R2),R3         ;GET THE WAS DATA
2797  003454  020304                            CMP     R3,R4           ;DID BIT GET CLEARED OK ??
2798  003456  001723                            BEQ     2$              ;BR IF YES
2799
2800  003460  004767  017206                    JSR     PC,SUER2        ;GO SET UP ERROR INFO
2801  003464  104002                            ERROR   2               ;FAILED TO CLR SELECTED BIT IN MAINT MODE
2802  003466  000717                            BR      2$              ;GO SELECT NEXT BIT FOR TEST
```

```
2805                                    ;*****************************************************************
 (3)                                    ;*TEST 6        TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
 (3)                                    ;*****************************************************************
 (2)  003470  000004                    TST6:   SCOPE
 (1)  003472  012767  000006  1755.14           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2806                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT ALL R/W BITS IN THE "LPR" REG-
 (1)                                    ISTER CAN BE SET AND CLEARED INDIVIDUALLY. A BIT MASK (RGMSK3:  177767)
 (1)                                    IS USED TO DEFINE THE BITS TO BE TESTED (ALL BUT BIT03). THE TEST
 (1)                                    SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A BIT TO TEST
 (1)                                            2. SET THE BIT AND VERIFY IT SET
 (1)                                            3. CLEAR THE BIT AND VERIFY IT CLEARED
 (1)                                            4. REPEAT 1 THRU 3 UNTIL ALL BITS TESTED
 (1)
 (1)                                    ANY ERRORS DETECTED ARE REPORTED AND AFTER THE ERROR, THE TEST
 (1)                                    RESUMES WITH THE NEXT BIT IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  4] IS CALLED TO REPORT BOTH FAIL TO SET AND FAIL TO
 (1)                                        CLEAR FAULTS.
 (1)
 (1)                                    SYNC:
 (1)                                    *******
 (1)
 (1)                                    1.) FAIL TO SET:        M7277   SH4     LOAD SSR LOW BYTE H     CR1
 (1)                                    2.) FAIL TO CLEAR:      M7277   SH4     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) IF ALL BITS FAIL THE PROBLEM IS MOST LIKELY THE M7277 MODULE
 (1)                                        (LPR LOAD SIGNALS)
 (1)                                    2.) IF NOT THEN IT IS PROBABLY AN "LPR" REGISTER CHIP OR
 (1)                                        BAD OUTPUT DATA MUX CHIP, BOTH ON THE M7278 MODULE.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7277   SH4     LOAD LPR H              EP2
 (1)
 (1)                                            M7278   SH5     LPR <15:12> L   (E52)
 (1)                                                    SH6     LPR <11:08> L   (E37)
 (1)                                                    SH7     LPR <07:04> L   (E59)
 (1)                                                    SH8     LPR <03:00> L   (E61)
 (1)                                                    SH5,6,7,8       OUTPUT MUX CHIPS  (74151'S PIN 2)
 (1)
 (1)                                    %
```

```
2807  003500  012767  003534  175402          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
2808  003506  010102                          MOV     R1,R2           ;COPY IT IN R2
2809  003510  062702  000004                  ADD     #LPR,R2         ;GENERATE REGADR IN R2
2810  003514  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
2811  003520  030567  021240          1$:     BIT     R5,RGMSK3       ;TEST THIS BIT ??
2812  003524  001003                          BNE     3$              ;BR IF YES
2813  003526  006305          2$:             ASL     R5              ;SHIFT THE MARKER
2814  003530  001430                          BEQ     TST7            ;;BR IF DONE ALL BITS
2815  003532  000772                          BR      1$              ;GO TEST NXT BIT
2816
2817  003534  010504          3$:             MOV     R5,R4           ;SET UP S/B DATA
2818  003536  005012                          CLR     (R2)            ;INIT REG BEING TESTED
2819  003540  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
2820  003546  010512                          MOV     R5,(R2)         ;SET LPR BIT
2821  003550  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2822  003552  020304                          CMP     R3,R4           ;DID IT SET
2823  003554  001403                          BEQ     4$              ;BR IF IT SET PROPERLY
2824
2825  003556  004767  017110                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
2826  003562  104004                          ERROR   4               ;LPR BIT FAILED TO SET PROPERLY
2827
2828  003564  005004          4$:             CLR     R4              ;GET READY TO CLEAR SELECTED BIT
2829  003566  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2830  003574  040512                          BIC     R5,(R2)         ;CLEAR THE BIT
2831  003576  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2832  003600  001752                          BEQ     2$              ;BR IF BIT CLEARED PROPERLY
2833
2834  003602  004767  017064                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
2835  003606  104004                          ERROR   4               ;LPR BIT FAILED TO CLEAR PROPERLY
2836  003610  000746                          BR      2$              ;GO SELECT NEXT BIT
```

```
2839                                    ;*****************************************************************
 (3)                                    ;*TEST 7        TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
 (3)                                    ;*****************************************************************
 (2)  003612  000004                    TST7:   SCOPE
 (1)  003614  012767  000007  175412            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2840                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT ALL BITS IN THE BREAK CONTROL REGISTER
 (1)                                    CAN BE SET AND CLEARED INDIVIDUALLY. IT USES A BIT MASK (RGMSK4:  177777)
 (1)                                    TO DEFINE THE R/W BITS (ALL 16.). R5 ALWAYS CONTAINS THE BIT CURRENTLY
 (1)                                    SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A BIT TO TEST
 (1)                                            2. SET THE BIT AND VERIFY THAT IT SET PROPERLY
 (1)                                            3. CLEAR THE BIT AND VERIFY THAT IT CLEARED PROPERLY
 (1)                                            4. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED.
 (1)
 (1)                                    ANY ERROR DETECTED IS REPORTED AND THE TEST RESUMES WITH THE NEXT BIT
 (1)                                    IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  5] IS CALLED TO REPORT BOTH FAIL TO SET PROPERLY AND FAIL
 (1)                                        TO CLEAR PROPERLY FAULTS.
 (1)
 (1)                                    SYNC:
 (1)                                    ******
 (1)
 (1)                                    1.) FAIL TO SET:        M7277   SH4     LOAD SSR LOW BYTE H     CR1
 (1)                                    2.) FAIL TO CLR:        M7277   SH4     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) THE ONLY DIFFERENCES IN THE DATA PATH HERE AND THAT FOT THE PREVIOUS
 (1)                                        TESTS ARE THE ACTUAL REGISTER CHIPS AND THE INPUT SELECTED ON THE
 (1)                                        OUTPUT DATA MULTIPLEXORS.
 (1)
 (1)                                    2.) IF ALL BITS FAIL THE PROBLEM IS MOST LIKELY THE M7277.
 (1)
 (1)                                    3.) IF ONLY ONE OR TWO FAIL THE PROBLEM IS MOST LIKELY THE M7278.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7277   SH4     LOAD BCR H              FU1
 (1)                                                            DATA TO BUS H           EN2
 (1)                                                            DATA SOURCE (A,B,C) H   DU1,DU2,DT2
 (1)
 (1)                                            M7278   SH5 - SH8 74175 REGISTER CHIPS (E51,E38,E67,E60)
 (1)                                                    SH5 - SH8 74151'S MUX CHIPS INPUT PIN 13
```

```
 (1)                                    %
2841  003622  012767  003656  175260            MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
2842  003630  010102                            MOV     R1,R2           ;GENERATE "BKR" ADDRESS IN R2
2843  003632  062702  000014                    ADD     #BKR,R2
2844  003636  012705  000001                    MOV     #1,R5           ;INIT BIT TEST MARKER
2845  003642  030567  021120            1$:     BIT     R5,RGMSK4       ;TEST THIS BIT ??
2846  003646  001003                            BNE     3$              ;BR IF YES
2847  003650  006305                    2$:     ASL     R5              ;SHIFT BIT MARKER
2848  003652  001430                            BEQ     TST10           ;;BR IF ALL BITS TESTED
2849  003654  000772                            BR      1$              ;GO TEST THE BIT
2850
2851  003656  010504                    3$:     MOV     R5,R4           ;SET UP S/B DATA
2852  003660  005012                            CLR     (R2)            ;INIT REG BEING TESTED
2853  003662  112761  000000  000016            MOVB    #0,SSR(R1)      ;SCOPE SYNC
2854  003670  050512                            BIS     R5,(R2)         ;SET THE SELECTED BIT IN "BKR"
2855  003672  011203                            MOV     (R2),R3         ;GET THE WAS DATA
2856  003674  020304                            CMP     R3,R4           ;DID BIT SET OK
2857  003676  001403                            BEQ     4$              ;BR IF YES
2858
2859  003700  004767  016766                    JSR     PC,SUER2        ;GO SET UP ERROR INFO
2860  003704  104005                            ERROR   5               ;BKR BIT FAILED TO SET PROPERLY
2861
2862  003706  005004                    4$:     CLR     R4              ;SET UP S/B DATA
2863  003710  112761  000000  000017            MOVB    #0,SSR+1(R1)    ;SCOPEE SYNC
2864  003716  040512                            BIC     R5,(R2)         ;CLEAR BKR BIT
2865  003720  011203                            MOV     (R2),R3         ;GET THE BKR WAS DATA
2866  003722  001752                            BEQ     2$              ;BR IF BKR BIT CLEARED OK
2867
2868  003724  004767  016742                    JSR     PC,SUER2        ;GO SET UP ERROR INFO
2869  003730  104005                            ERROR   5               ;BKR BIT FAILED TO CLR PROPERLY
2870  003732  000746                            BR      2$              ;GO SELECT NEXT BIT
```

```
                                ;*****************************************************************
                                ;*TEST 10       TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
                                ;*****************************************************************
003734  000004                  TST10:  SCOPE
003736  012767  000010  175270          MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                        .REM    %
                        TEST ABSTRACT:
                        **************

                                THIS TEST VERIFIES THAT ALL R/W BITS IN THE SILO STATUS REGISTER
                        (SSR) CAN BE SET AND CLEARED INDIVIDUALLY. IT USES A BIT MASK (RGMSK5:  100077)
                        TO DEFINE THE R/W BITS (15,5,4,3,2,1, AND 0). RS ALWAYS CONTAINS THE BIT
                        CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                1. SELECT A BIT TO TEST
                                2. SET THE BIT AND VERIFY THAT IT SET PROPERLY
                                3. CLEAR THE BIT AND VERIFY THAT IT CLEARED PROPERLY
                                4. REPEAT 1 THRU 3 UNTIL ALL BITS ARE TESTED

                        ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES WITH THE
                        NEXT BIT IN SEQUENCE.

                        ERRORS:
                        ********

                        1.) [ERROR  6] IS CALLED TO REPOORT BOTH FAIL TO SET PROPERLY AND
                            FAIL TO CLEAR PROPERLY FAULTS.

                        SYNC:
                        *****

                        1.) FAIL TO SET:  M7277 SH4     LOAD LPR H      EP2
                        2.) FAIL TO CLR:  M7277 SH4     LOAD BCR H      FU1

                        DEBUG:
                        ******

                        1.) THE ONLY DIFFERENCES BETWEEN THEE DATA PATHS USED BY THIS TEST AND
                            THAT USED BY THE PREVIOUS TESTS ARE THE ACTUAL "SSR" REGISTER
                            CHIPS AND THE INPUT PIN SELECTED ON THE OUTPUT DATA MULTIPLEXORS.
                        2.) IF ALL BITS FAIL IT IS MOST LIKELY THE M7277
                        3.) IF BITS <13:08> FAIL IT IS MOST LIKELY THE M7279
                        4.) IF JUST ONE OR TWO BITS FAIL IT IS MOST LIKELY THE M7278

                        KEY LOGIC:
                        **********

                                M7277   SH4     LOAD SSR LOW BYTE H     CR1
                                                LOAD SSR HIGH BYTE H    CP2
                                                DATA TO BUS H           EN2
                                                DATA SOURCE (A,B,C) H   DU1,DU2,DT2

                                M7279   SH2     SSR <13:08> H  (E20 AND E24)
```

```
(1)                                                 M7278   SH5 - SH8       REGISTER CHIPS E53,E68, OR E69 (74175'S)
(1)                                                                         OUTPUT MUX CHIPS - (74151'S  PIN 12)
(1)                                         %
2875  003744  012767  004000  175136          MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
2876  003752  010102                          MOV     R1,R2           ;GENERATE "SSR" ADDRESS IN R2
2877  003754  062702  000016                  ADD     #SSR,R2
2878  003760  012705  000001                  MOV     #1,R5           ;INIT BIT TEST MARKER
2879  003764  030567  021000          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
2880  003770  001003                          BNE     3$              ;BR IF YES
2881  003772  006305                  2$:     ASL     R5              ;SHIFT BIT MARKER
2882  003774  001435                          BEQ     TST11           ;BR IF ALL BITS TESTED
2883  003776  000772                          BR      1$              ;GO TEST THE BIT
2884
2885  004000  010504                  3$:     MOV     R5,R4           ;SET UP S/B DATA
2886  004002  005012                          CLR     (R2)            ;INIT REG BEING TESTED
2887  004004  012761  000000  000004          MOV     #0,LPR(R1)      ;SCOPE SYNC
2888  004012  050512                          BIS     R5,(R2)         ;SET THE SELECTED BIT IN "SSR"
2889  004014  011203                          MOV     (R2),R3         ;GET THE WAS DATA
2890  004016  042703  077700                  BIC     #77700,R3       ;CLEAR OUT DON'T CARE BITS
2891  004022  020304                          CMP     R3,R4           ;DID BIT SET OK
2892  004024  001403                          BEQ     4$              ;BR IF YES
2893
2894  004026  004767  016640                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
2895  004032  104006                          ERROR   6               ;SSR BIT FAILED TO SET PROPERLY
2896
2897  004034  005004                  4$:     CLR     R4              ;SET UP S/B DATA
2898  004036  01276?  000000  000014          MOV     #0,BKR(R1)      ;SCOPE SYNC
2899  004044  0405??                          BIC     R5,(R2)         ;CLEAR SSR BIT
2900  004046  0112??                          MOV     (R2),R3         ;GET THE SSR WAS DATA
2901  004050  042703  077700                  BIC     #77700,R3       ;CLEAR JUNK BITS
2902  004054  020304                          CMP     R3,R4           ;DID THE SSR BIT GET CLEARED ??
2903  004056  00??45                          BEQ     2$              ;BR IF SSR BIT CLEARED OK
2904
2905  004060  004767  016606                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
2906  004064  104006                          ERROR   6               ;SSR BIT FAILED TO CLR PROPERLY
2907  004066  000741                          BR      2$              ;GO SELECT NEXT BIT
```

```
2910                                    ;*****************************************************************
 (3)                                    ;*TEST 11       TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
 (3)                                    ;*****************************************************************
 (2)  004070  000004                    TST11:  SCOPE
 (1)  004072  012767  000011  175134            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2911                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT SETTING AND CLEARING EACH R/W BIT IN
 (1)                                    THE "LPR" REGISTER DOES NOT DISTURB (CLEAR) ANY OTHER BIT IN THE REGISTER.
 (1)                                    A BIT MASK (RGMSK3:  177767) IS USED TO DEFINE THE R/W BITS (ALL BUT BIT 03).
 (1)                                    R5 ALWAYS CONTAINS THE BIT CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE
 (1)                                    IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A BIT TO TEST
 (1)                                            2. SET ALL THE WRITABLE BITS
 (1)                                            3. CLEAR THE SELECTED BIT - VERIFY IT CLEARED PROPERLY
 (1)                                            4. SET THE SELECTED BIT - VERIFY IT SET PROPERLY
 (1)                                            5. REPEAT 1 THRU 4 UNTIL ALL BITS ARE TESTED
 (1)
 (1)                                    ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES WITH THE
 (1)                                    NEXT BIT IN SEQUENCE .
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  41 IS CALLED TO REPORT BOTH FAIL TO CLEAR PROPERLY AND
 (1)                                        FAIL TO SET PROPERLY FAULTS.
 (1)
 (1)                                    SYNC:
 (1)                                    *******
 (1)
 (1)                                    1.) FAIL TO CLR:        M7277   LOAD SSR LOW BYTE H     CR1
 (1)                                    2.) FAIL TO SET:        M7277   LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) PROBLEMS DETECTED BY THIS TEST INDICATE ADJACENT BIT INTERFERENCE
 (1)                                        CAUSED BY CROSS TALK OR NOISE. PROBLEM IS MOST LIKELY THE M7278.
 (1)
 (1)                                    KEY LOGIC:      (SAME AS FOR TEST 6)
 (1)                                    ***********
 (1)                                    %
2912  004100  012767  004134  175002            MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
2913  004106  010102                            MOV     R1,R2           ;SET UP THE REG ADDR
2914  004110  062702  000004                    ADD     #LPR,R2
2915  004114  012705  000001                    MOV     #1,R5           ;INIT BIT TEST MASK
2916  004120  030567  020640           1$:      BIT     R5,RGMSK3       ;TEST THIS BIT ??
2917  004124  001003                            BNE     3$              ;BR IF YES
2918  004126  006305                    2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
2919  004130  001436                            BEQ     TST12           ;;BR IF TESTED ALL BITS
2920  004132  000772                            BR      1$              ;GO TEST THIS BIT
```

```
2921
2922  004134  016704  020624          3$:   MOV     RGMSK3,R4       ;SET UP S/B DATA
2923  004140  005012                        CLR     (R2)            ;INIT REG BEING TESTED
2924  004142  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
2925  004150  040504                        BIC     R5,R4           ;CLR BIT "N"
2926  004152  016712  020606                MOV     RGMSK3,(R2)     ;SET ALL R/W BITS IN LPR
2927  004156  040512                        BIC     R5,(R2)         ;CLEAR BIT "N" IN LPR
2928  004160  011203                        MOV     (R2),R3         ;GET THE WAS DATA
2929  004162  020304                        CMP     R3,R4           ;DID IT CLEAR OK ?
2930  004164  001404                        BEQ     4$              ;BR IF YES
2931
2932  004166  004767  016500                JSR     PC,SUER2        ;GO SET UP ERROR INFO
2933  004172  104004                        ERROR   4               ;BIT "N" FAILED TO CLR PROPERLY
2934  004174  000754                        BR      2$              ;GO TEST NEXT BIT
2935
2936  004176  050504                4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
2937  004200  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2938  004206  050512                        BIS     R5,(R2)         ;SET BIT "N" IN LPR
2939  004210  011203                        MOV     (R2),R3         ;GET THE WAS DATA
2940  004212  020304                        CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
2941  004214  001744                        BEQ     2$              ;BR IF YES
2942
2943  004216  004767  016450                JSR     PC,SUER2        ;GO SET UP ERROR INFO
2944  004222  104004                        ERROR   4               ;BIT "N" FAILED TO SET PROPERLY
2945  004224  000740                        BR      2$              ;GO SELECT NEXT BIT
```

```
2948                                    ;*****************************************************************
 (3)                                    ;*TEST 12       TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS
 (3)                                    ;*****************************************************************
 (2)  004226  000004                    TST12:  SCOPE
 (1)  004230  012767  000012  174776            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
2949                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT CLEARING AND SETTING E'CH R/W
 (1)                                    BIT IN THE BREAK CONTROL REGISTER INDIVIDUALLY DOES NOT DISTURB ANY
 (1)                                    OF THE OTHER BITS. A BIT MASK (RGMSK4:  177777) IS USED TO DEFINE
 (1)                                    THE R/W BITS (ALL 16.). R5 ALWAYS CONTAINS THE BIT CURRENTLY SELECTED
 (1)                                    FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A BIT TO TEST
 (1)                                            2. SET ALL WRITABLE BITS IN THE "BKR"
 (1)                                            3. CLEAR THE SELECTED BIT AND VERIFY THAT IT CLEARED PROPERLY
 (1)                                            4. SET THE SELECTED BIT AND VERIFY THAT IT SET PROPERLY
 (1)                                            5. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED
 (1)
 (1)                                    ANY ERROR DETECTED IS REORTRD AND THEN THE TEST RESUMES WITH THE
 (1)                                    NEXT BIT IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  5] IS CALLED TO REPORT BOTH CLEAR AND SET FAULTS.
 (1)
 (1)                                    SYNC:
 (1)                                    *****
 (1)
 (1)                                    1.) FAIL TO CLR:        M7277   SM4     LOAD SSR LOW BYTE H     CR1
 (1)                                    2.) FAIL TO SET:        M7277   SM4     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) LIKE THE PREVIOUS TEST, FAILURES HERE INDICATE ADJACENT BIT
 (1)                                        INTERFERENCE CAUSED BY CROSS TALK OR NOISE. THE FAULT IS MOST
 (1)                                        LIKELY THE M7278 MODULE.
 (1)
 (1)                                    KEY LOGIC:      (SAME AS FOR TEST 7)
 (1)                                    **********
 (1)                                    %
2950  004236  012767  004272  174644            MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
2951  004244  010102                    1$:     MOV     R1,R2           ;SET UP THE REG ADDR
2952  004246  062702  000014                    ADD     #BKR,R2
2953  004252  012705  000001                    MOV     #1,R5           ;INIT BIT TEST MASK
2954  004256  030567  020504            2$:     BIT     R5,RGMSK4       ;TEST THIS BIT ??
2955  004262  001003                            BNE     3$              ;BR IF YES
2956  004264  006305                            ASL     R5              ;SHIFT THE BIT TEST MASK
2957  004266  001436                            BEQ     TST13           ;;BR IF TESTED ALL BITS
2958  004270  000772                            BR      1$              ;GO TEST THIS BIT
```

N12


```
2959
2960  004272  016704  020470          3$:   MOV     RGMSK4,R4       ;SET UP S/B DATA
2961  004276  040504                        BIC     R5,R4           ;CLR BIT "N"
2962  004300  005012                        CLR     (R2)            ;INIT REG BEING TESTED
2963  004302  016712  020460                MOV     RGMSK4,(R2)     ;SET ALL R/W BITS IN BKR
2964  004306  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
2965  004314  040512                        BIC     R5,(R2)         ;CLEAR BIT "N" IN BKR
2966  004316  011203                        MOV     (R2),R3         ;GET THE WRS DATA
2967  004320  020304                        CMP     R3,R4           ;DID IT CLEAR OK ?
2968  004322  001404                        BEQ     4$              ;BR IF YES
2969
2970  004324  004767  016342                JSR     PC,SUER2        ;GO SET UP ERROR INFO
2971  004330  104005                        ERROR   5               ;BIT "N" FAILED TO CLR PROPERLY
2972  004332  000754                        BR      2$              ;GO TEST NEXT BIT
2973
2974  004334  050504                  4$:   BIS     R5,R4           ;SET BIT "N" IN S/B DATA
2975  004336  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
2976  004344  050512                        BIS     R5,(R2)         ;SET BIT "N" IN BKR
2977  004346  011203                        MOV     (R2),R3         ;GET THE WRS DATA
2978  004350  020304                        CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
2979  004352  001744                        BEQ     2$              ;BR IF YES
2980
2981  004354  004767  016312                JSR     PC,SUER2        ;GO SET UP ERROR INFO
2982  004360  104005                        ERROR   5               ;BIT "N" FAILED TO SET PROPERLY
2983  004362  000740                        BR      2$              ;GO SELECT NEXT BIT
```

```
                                        ;*****************************************************************
                                        ;*TEST 13       TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
                                        ;*****************************************************************
004364  000004
004366  012767  000013  174640          TST13:  SCOPE
                                                MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THIS TEST VERIFIES THAT CLEARING AND SETTING EACH R/W
                                        BIT IN THE SILO STATUS REGISTER INDIVIDUALLY DOES NOT DISTURB ANY
                                        OF THE OTHER BITS. A BIT MASK (RGMSKS:  100077) IS USED TO DEFINE
                                        THE R/W BITS (15,5,4,3,2,1, AND 0). R5 ALWAYS CONTAINS THE BIT CURRENTLY SELECTED
                                        FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

                                                1. SELECT A BIT TO TEST
                                                2. SET ALL WRITABLE BITS IN THE "SSR"
                                                3. CLEAR THE SELECTED BIT AND VERIFY THAT IT CLEARED PROPERLY
                                                4. SET THE SELECTED BIT AND VERIFY THAT IT SET PROPERLY
                                                5. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED

                                        ANY ERROR DETECTED IS REORTRD AND THEN THE TEST RESUMES WITH THE
                                        NEXT BIT IN SEQUENCE.

                                        ERRORS:
                                        ********

                                        1.) [ERROR  6] IS CALLED TO REPORT BOTH CLEAR AND SET FAULTS.

                                        SYNC:
                                        *****

                                        1.) FAIL TO CLR:        M7277   SHH     LOAD LPR H      EP2
                                        2.) FAIL TO SET:        M7277   SHH     LOAD BCR H      FUI

                                        DEBUG:
                                        ******

                                        1.) LIKE THE PREVIOUS TEST, FAILURES HERE INDICATE ADJACENT BIT
                                            INTERFERENCE CAUSED BY CROSS TALK OR NOISE. THE FAULT IS MOST
                                            LIKELY THE M7278 MODULE.

                                        KEY LOGIC:      (SAME AS FOR TEST 10)
                                        *********
                                        %
004374  012767  004430  174506                  MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
004402  010102                                  MOV     R1,R2           ;SET UP THE REG ADDR
004404  062702  000016                          ADD     #SSR,R2
004410  012705  000001                          MOV     #1,R5           ;INIT BIT TEST MASK
004414  030567  020350          1$:     BIT     R5,RGMSK5       ;TEST THIS BIT ??
004420  001003                          BNE     3$              ;BR IF YES
004422  006305                  2$:     ASL     R5              ;SHIFT THE BIT TEST MASK
004424  001442                          BEQ     TST14           ;;BR IF TESTED ALL BITS
004426  000772                          BR      1$              ;GO TEST THIS BIT
```

```
2997
2998  004430  016704  020334          3$:   MOV     RGMSK5,R4       ;SET UP S/B DATA
2999  004434  040504                        BIC     R5,R4           ;CLR BIT "N"
3000  004436  005012                        CLR     (R2)            ;INIT REG BEING TESTED
3001  004440  016712  020324                MOV     RGMSK5,(R2)     ;SET ALL R/W BITS IN SSR
3002  004444  012761  000000  000004        MOV     #0,LPR(R1)      ;SCOPE SYNC
3003  004452  040512                        BIC     R5,(R2)         ;CLEAR BIT "N" IN SSR
3004  004454  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3005  004456  042703  077700                BIC     #77700,R3       ;CLEAR JUNK BITS
3006  004462  020304                        CMP     R3,R4           ;DID IT CLEAR OK ?
3007  004464  001404                        BEQ     4$              ;BR IF YES
3008
3009  004466  004767  016200                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3010  004472  104006                        ERROR   6               ;BIT "N" FAILED TO CLR PROPERLY
3011  004474  000752                        BR      2$              ;GO TEST NEXT BIT
3012
3013  004476  050504                4$:     BIS     R5,R4           ;SET BIT "N" IN S/B DATA
3014  004500  012761  000000  000014        MOV     #0,BKR(R1)      ;SCOPE SYNC
3015  004506  050512                        BIS     R5,(R2)         ;SET BIT "N" IN SSR
3016  004510  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3017  004512  042703  077700                BIC     #77700,R3       ;CLEAR JUNK BITS
3018  004516  020304                        CMP     R3,R4           ;DID BIT "N" SET PROPERLY ?
3019  004520  001740                        BEQ     2$              ;BR IF YES
3020
3021  004522  004767  016144                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3022  004526  104006                        ERROR   6               ;BIT "N" FAILED TO SET PROPERLY
3023  004530  000734                        BR      2$              ;GO SELECT NEXT BIT
```

```
3026                                    ;*****************************************************************
 (3)                                    ;*TEST 14        "CAR" MEMORY ADDRESSING TEST
 (3)                                    ;*****************************************************************
 (2)  004532  000004                    TST14:  SCOPE
 (1)  004534  012767  000014  174472            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3027                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT EACH LOCATION IN THE CURRENT ADDRESS
 (1)                                    MEMORY CAN BE UNIQUELY ADDRESSED. IT WRITES THE PATTERN SHOWN BELOW
 (1)                                    INTO THE MEMORY AND THEN READS BACK EACH LOCATION TO VERIFY THAT IT
 (1)                                    WAS WRITTEN CORRECTLY. SINCE THE MEMORY LOGIC IS PARTITIONED INTO
 (1)                                    FOUR 16 X 4 READ/WRITE MEMORY CHIPS, THE PATTERN RESULTS IN THE LINE
 (1)                                    NUMBER (00 - 17(8)) BEING WRITTEN AS DATA INTO TO EACH LOCATION
 (1)                                    (00 - 17(8)) IN EACH CHIP. THAT IS:  LOC00 = 00, LOC 01 = 01,
 (1)                                    .......LOC 17 = 17.
 (1)
 (1)
 (1)                                            MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)
 (1)
 (1)                                                            00              000000
 (1)                                                            01              010421
 (1)                                                            02              021042
 (1)                                                            03              031463
 (1)                                                            04              042104
 (1)                                                            05              052525
 (1)                                                            06              063146
 (1)                                                            07              073567
 (1)                                                            10              104210
 (1)                                                    11                              114631
 (1)                                                            12              125252
 (1)                                                            13              135673
 (1)                                                            14              146314
 (1)                                                            15              156735
 (1)                                                            16              167356
 (1)                                                            17              177777
 (1)
 (1)                                    ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES CHECKING
 (1)                                    THEE NEXT LOCATION IN SEQUENCE UNTIL ALL 16. HAVE BEEN CHECKED.
 (1)
 (1)                                            NOTE:   THIS TEST ALWAYS CHECKS ALL 16. LINES REGARDLESS
 (1)                                                    OF HOW THE LINE SELECTION PARAMETER WAS INITIALLY
 (1)                                                    SET UP.
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  7] IS CALLED TO REPORT ANY LINES (LOCATIONS) THAT FAIL.
 (1)                                        THE FAILING LINE # IS INCLUDED AS PART OF THE ERROR HEADER MESSAGE.
 (1)
 (1)                                    SYNC:
 (1)                                    ******
 (1)
```

```
(1)                                 1.) WRITE SYNC: M7277   SH4     LOAD SSR LOW BYTE H     CR1
(1)                                 2.) READ SYNC:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
(1)
(1)                                 DEBUG:
(1)                                 ******
(1)
(1)                                 1.) ANALYZE THE ERROR REPORTS CAREFULLY ASKING THE FOLLOWING QUESTIONS:
(1)
(1)                                       A. DOES THE FAULT AFFECT ONLY ONE LINE ?
(1)                                       B. DOES THE FAULT AFFECT ONLY ONE 4-BIT DATA GROUP ?
(1)                                          IE <15:12>, <11:08>, <07:04>, OR <03:00>
(1)                                       C. DOES ANY DATA AT ALL APPEAR TO BE WRITTEN ?
(1)
(1)                                 2.) IF "A" IS TRUE THEN SUSPECT AN ADDRESSING PROBLEM IN THE MEMORY
(1)                                     ADDRESS MUX.
(1)
(1)                                 3.) IF "B" IS TRUE THEN SUSPECT A DATA MUX, UP-COUNTER, MEMORY,
(1)                                     OR INVERTER CHIP PROBLEM.
(1)
(1)                                 4.) IF "C" IS TRUE SUSPECT A MEMORY WRITE TIMING PROBLEM.
(1)
(1)                                 5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7278.
(1)
(1)                                 KEY LOGIC:
(1)                                 **********
(1)
(1)                                       M7277   SH4     LOAD CA H               E58-13
(1)                                                       DATA TO BUS H          EN2
(1)                                                       DATA SOURCE (A,B,C)    DU1,DU2,DT2
(1)
(1)                                               SH5     MEMADD SOURCE SEL H    E55-8
(1)                                                       CA MEM WRITE ENAB L    E50-1
(1)                                                       BUF ADDRS TO BUS H     E33-1 (SHD BE LOW)
(1)                                                       74157 MUX CHIPS E33,E27,E20 BITS<17:08>
(1)                                                       74193 COUNTER CHIPS E19,E26,E32 BITS<17:08>
(1)                                                       7489 MEMORY CHIPS E18,E25,E31 BITS<17:08>
(1)                                                       7404 INVERTER CHIPS E30,E24,E17 BITS<17:08>
(1)
(1)                                               SH5     74157 MUX CHIP  E48
(1)                                                       74157 DATA MUX CHIPS E13,E06 BITS<07:00>
(1)                                                       74193 COUNTER CHIPS E12,E05 BITS<07:00>
(1)                                                       7489 MEMORY CHIPS E11,E04 BITS<07:00>
(1)
(1)
(1)                                       M7278   SH5 THRU SH8    74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)
(1)                                 %
3028  004542  010102                  MOV     R1,R2               ;COPY IT IN R2
3029  004544  062702  000006          ADD     #CAR,R2             ;SET UP REGADR IN R2
3030  004550  016767  017634  174440  MOV     LINSEL,$TMP7        ;SAVE LINE SELECT PARAMETER
3031  004556  012767  177777  017624  MOV     #-1,LINSEL          ;DO ALL LINES FOR THIS TEST
3032  004564  004767  016234      1$: JSR     PC,SELINE           ;GO SELECT A LINE NO.
3033  004570  000415                  BR      3$                  ;;BR IF DONE ALL SELECTED LINES
3034  004572  105767  020346          TSTB    LINE                ;DOING LINE 00 ?
3035  004576  001001                  BNE     2$                  ;BR IF NOT
```

```
3036  004600  005004                          CLR     R4              ;INIT TEST DATA
3037
3038  004602  156711  020336          2$:     BISB    LINE,(R1)       ;SELECT A LINE
3039  004606  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
3040  004614  010412                          MOV     R4,(R2)         ;LOAD THE CAR REG.
3041  004616  062704  010421                  ADD     #10421,R4       ;GENERATE NEW DATA
3042  004622  000760                          BR      1$              ;GO DO NEXT LINE
3043
3044  004624  004767  016174          3$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
3045  004630  000434                          BR      7$              ;BR IF CHECKED ALL LINES
3046  004632  105767  020306                  TSTB    LINE            ;DOING LINE 00 ?
3047  004636  001001                          BNE     4$              ;BR IF NOT
3048  004640  005004                          CLR     R4              ;INIT S/B DATA
3049
3050  004642  156711  020276          4$:     BISB    LINE,(R1)       ;SELECT A LINE
3051  004646  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3052  004654  011203                          MOV     (R2),R3         ;GET CONTENTS OF CAR
3053  004656  020304                          CMP     R3,R4           ;WAS DATA OK ?
3054  004660  001412                          BEQ     5$              ;BR IF YES
3055
3056  004662  004767  016004                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
3057  004666  004567  016224                  JSR     R5,SUNUM        ;SET UP LINE NO. IN MSG BUFFER
3058  004672  025144                          LINE
3059  004674  025777                          EM7+47
3060  004676  012767  004714  174204          MOV     #6$,SLPERR      ;SET UP ERROR LOOP RETURN
3061  004704  104007                          ERROR   7               ;CAR ADDRESSING ERROR
3062
3063  004706  062704  010421          5$:     ADD     #10421,R4       ;GENERATE NEW S/B DATA
3064  004712  000744                          BR      3$              ;GO CHECK NEXT LINE
3065
3066  004714  005067  020224          6$:     CLR     LINE            ;RESTART AT LINE 00 IF LOOPING
3067  004720  000721                          BR      1$              ;GO RESTART
3068
3069  004722  016767  174270  017460  7$:     MOV     STMP7,LINSEL    ;RESTORE LINE SELECT PARAMETER
```

```
3072                                    ;*********************************************************************
 (3)                                    ;*TEST 15       "BCR" MEMORY ADDRESSING TEST
 (3)                                    ;*********************************************************************
 (2)  004730  000004                    TST15:  SCOPE
 (1)  004732  012767  000015  174274            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3073                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT EACH LOCATION IN THE BYTE COUNT
 (1)                                    MEMORY CAN BE UNIQUELY ADDRESSED. IT WRITES THE PATTERN SHOWN BELOW
 (1)                                    INTO THE MEMORY AND THEN READS BACK EACH LOCATION TO VERIFY THAT IT
 (1)                                    WAS WRITTEN CORRECTLY. SINCE THE MEMORY LOGIC IS PARTITIONED INTO
 (1)                                    FOUR 16 X 4 READ/WRITE MEMORY CHIPS, THE PATTERN RESULTS IN THE LINE
 (1)                                    NUMBER (00 - 17(8)) BEING WRITTEN AS DATA INTO TO EACH LOCATION
 (1)                                    (00 - 17(8)) IN EACH CHIP. THAT IS:  LOC00 = 00, LOC 01 = 01,
 (1)                                    .......LOC 17 = 17.
 (1)
 (1)
 (1)                                            MEMORY PATTERN: LOCATION        CONTENTS        (BOTH OCTAL)
 (1)
 (1)                                                            00              000000
 (1)                                                            01              010421
 (1)                                                            02              021042
 (1)                                                            03              031463
 (1)                                                            04              042104
 (1)                                                            05              052525
 (1)                                                            06              063146
 (1)                                                            07              073567
 (1)                                                            10              104210
 (1)                                                            11              114631
 (1)                                                            12              125252
 (1)                                                            13              135673
 (1)                                                            14              146314
 (1)                                                            15              156735
 (1)                                                            16              167356
 (1)                                                            17              177777
 (1)
 (1)                                    ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES CHECKING
 (1)                                    THE NEXT LOCATION IN SEQUENCE UNTIL ALL 16. HAVE BEEN CHECKED.
 (1)
 (1)                                            NOTE:   THIS TEST ALWAYS CHECKS ALL 16. LINES REGARDLESS
 (1)                                                    OF HOW THE LINE SELECTION PARAMETER WAS INITIALLY
 (1)                                                    SET UP.
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  10] IS CALLED TO REPORT ANY LINES (LOCATIONS) THAT FAIL.
 (1)                                        THE FAILING LINE # IS INCLUDED AS PART OF THE ERROR HEADER MESSAGE.
 (1)
 (1)                                    SYNC:
 (1)                                    ******
 (1)
```

```
(1)                                 1.) WRITE SYNC: M7277   SH4     LOAD SSR LOW BYTE H     CR1
(1)                                 2.) READ SYNC:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
(1)
(1)                                 DEBUG:
(1)                                 ******
(1)
(1)                                 1.) ANALYZE THE ERROR REPORTS CAREFULLY ASKING THE FOLLOWING QUESTIONS:
(1)
(1)                                       A. DOES THE FAULT AFFECT ONLY ONE LINE ?
(1)                                       B. DOES THE FAULT AFFECT ONLY ONE 4-BIT DATA GROUP ?
(1)                                          IE <15:12>, <11:08>, <07:04>, OR <03:00>
(1)                                       C. DOES ANY DATA AT ALL APPEAR TO BE WRITTEN ?
(1)
(1)                                 2.) IF "A" IS TRUE THEN SUSPECT AN ADDRESSING PROBLEM IN THE MEMORY
(1)                                     ADDRESS MUX.
(1)
(1)                                 3.) IF "B" IS TRUE THEN SUSPECT A DATA MUX, UP-COUNTER, MEMORY,
(1)                                     OR INVERTER CHIP PROBLEM.
(1)
(1)                                 4.) IF "C" IS TRUE SUSPECT A MEMORY WRITE TIMING PROBLEM.
(1)
(1)                                 5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7278.
(1)
(1)                                 KEY LOGIC:
(1)                                 **********
(1)
(1)                                       M7277   SH4     LOAD BC H               FU2
(1)                                                       DATA TO BUS H          EN2
(1)                                                       DATA SOURCE (A,B,C)    DU1,DU2,DT2
(1)
(1)                                       SH5     MEMADD SOURCE SEL H    E55-8
(1)                                               BC MEM WRITE ENAB L    E57-4
(1)                                               BUF ADDRS TO BUS H     E33-1 (SHD BE LOW)
(1)                                       M7278   SH3     BITS<15:08>
(1)                                               74157 INPUT MUX CHIPS E18,E19
(1)                                               74193 UP COUNTER CHIPS E27,E26
(1)                                               7489 MEMORY CHIPS E33,E34
(1)                                               7404 INVERTER CHIPS E41,E42
(1)
(1)                                       SH4     BITS<07:00>
(1)                                               74157 INPUT MUX CHIPS E16,E17
(1)                                               74193 UP COUNTER CHIPS E24,E25
(1)                                               7489 MEMORY CHIPS E31,E32
(1)                                               7404 INVERTER CHIPS E39,E40
(1)
(1)                                       SH5 THRU SH8    74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)
(1)                               %
3074  004740  010102                    MOV     R1,R2           ;COPY IT IN R2
3075  004742  062702  000010            ADD     #BCR,R2         ;SET UP REGADR IN R2
3076  004746  016767  017436  174242    MOV     LINSEL,$TMP7    ;SAVE LINE SELECT PARAMETER
3077  004754  012767  177777  017426    MOV     #-1,LINSEL      ;DO ALL LINES FOR THIS TEST
3078  004762  004767  016036        1$: JSR     PC,SELINE       ;GO SELECT A LINE NO.
3079  004766  000415                    BR      3$              ;;BR IF DONE ALL SELECTED LINES
3080  004770  105767  020150            TSTB    LINE            ;DOING LINE 00 ?
```

```
3081  004774  001001                        BNE     2$              ;BR IF NOT
3082  004776  005004                        CLR     R4              ;INIT TEST DATA
3083
3084  005000  156711  020140          2$:   BISB    LINE,(R1)       ;SELECT A LINE
3085  005004  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
3086  005012  010412                        MOV     R4,(R2)         ;LOAD THE BCR REG.
3087  005014  062704  01042.                ADD     #10421,R4       ;GENERATE NEW DATA
3088  005020  000760                        BR      1$              ;GO DO NEXT LINE
3089
3090  005022  004767  015776          3$:   JSR     PC,SELINE       ;GO SELECT A LINE NO.
3091  005026  000434                        BR      7$              ;;BR IF CHECKED ALL LINES
3092  005030  105767  020110                TSTB    LINE            ;DOING LINE 00 ?
3093  005034  001001                        BNE     4$              ;BR IF NOT
3094  005036  005004                        CLR     R4              ;INIT S/B DATA
3095
3096  005040  156711  020100          4$:   BISB    LINE,(R1)       ;SELECT A LINE
3097  005044  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3098  005052  011203                        MOV     (R2),R3         ;GET CONTENTS OF BCR
3099  005054  020304                        CMP     R3,R4           ;WAS DATA OK ?
3100  005056  001412                        BEQ     5$              ;BR IF YES
3101
3102  005060  004767  015606                JSR     PC,SUER2        ;GO SET UP ERROR INFO
3103  005064  004567  016026                JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
3104  005070  025144                        LINE
3105  005072  026046                        EM10+44
3106  005074  012767  005112  174006        MOV     #6$,$LPERR      ;SET UP ERROR LOOP RETURN
3107  005102  104010                        ERROR   10              ;BCR ADDRESSING ERROR
3108
3109  005104  062704  010421          5$:   ADD     #10421,R4       ;GENERATE NEW S/B DATA
3110  005110  000744                        BR      3$              ;GO CHECK NEXT LINE
3111
3112  005112  005067  020026          6$:   CLR     LINE            ;RESTART AT LINE 00 IF LOOPING
3113  005116  000721                        BR      1$              ;GO RESTART
3114
3115  005120  016767  174C 2  017262  7$:   MOV     $TMP7,LINSEL    ;RESTORE THE LINE SELECT PARAMETER
```

```
3118                                    ;*************************************************************
 (3)                                    ;*TEST 16       "CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
 (3)                                    ;*************************************************************
 (2) 005126  000004                     TST16:  SCOPE
 (1) 005130  012767  000016  174076             MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3119                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THE ABILITY TO SET AND CLEAR ALL BITS IN
 (1)                                    ALL THE SELECTED LOCATIONS (LINES) OF THE CURRENT ADDRESS MEMORY.
 (1)                                    IT USES THE CONFIGURATION PARAMETER (LINSEL:) TO DEFINE WHICH LINES TO
 (1)                                    TEST. THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A LINE # TO TEST
 (1)                                            2. LOAD THE SELECTED LOCATION WITH 177777
 (1)                                            3. READ IT BACK TO VERIFY ALL BITS SET
 (1)                                            4. LOAD THE SELECTED LOCATION WITH 000000
 (1)                                            5. READ IT BACK TO VERIFY ALL BITS CLEARED
 (1)                                            6. REPEAT STEPS 1 THRU 5 UNTIL ALL SELECTED LINES ARE TESTED.
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT
 (1)                                    LINE # IN SEQUENCE AS DEFINED BY "LINSEL".
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  7] IS CALLED TO REPORT ALL DATA COMPARE ERRORS
 (1)
 (1)                                    SYNC:
 (1)                                    *****
 (1)
 (1)                                    1.) WRITE 1'S:  M7277   SH4     LOAD SSR LOW BYTE H     CR1
 (1)
 (1)                                    2.) WRITE 0'S:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:  (REFER TO TEST 14)
 (1)                                    ******
 (1)
 (1)                                    KEY LOGIC:      (REFER TO TEST 14)
 (1)                                    **********
 (1)
 (1)                                    %
3120  005136  010102                            MOV     R1,R2           ;COPY IT INTO R2
3121  005140  062702  000006                    ADD     #CAR,R2         ;R2 GETS CAR ADDRESS
3122  005144  004767  015654           1$:      JSR     PC,SELINE       ;GO SELECT A LINE NO.
3123  005150  000443                            BR      TST17           ;;BR IF DONE ALL SELECTED LINES
3124  005152  012704  177777                    MOV     #-1,R4          ;RESULT IN CAR S/B = 177777
3125  005156  156711  017762                    BISB    LINE,(R1)       ;SELECT A LINE NO.
3126  005162  004567  015730                    JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
3127  005166  025144                            LINE
3128  005170  025777                            EM7+47
3129
3130  005172  112761  000000  000016  2$:      MOVB    #0,SSR(R1)      ;SCOPE SYNC
```

```
3131  005200  010412                          MOV     R4,(R2)         ;LOAD A CAR WITH ALL ONES
3132  005202  011203                          MOV     (R2),R3         ;GET THE WAS DATA FROM THE CAR
3133  005204  020403                          CMP     R4,R3           ;DID IT CONTAIN ALL ONES ?
3134  005206  001406                          BEQ     3$              ;;BR IF ALL 1'S
3135
3136  005210  004767  015456                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
3137  005214  012767  005172  173666          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3138  005222  104007                          ERROR   7               ;FAILED TO SET ALL 1'S IN SELECTED CAR
3139
3140  005224  005004                  3$:     CLR     R4              ;RESULT IN CAR S/B = 000000
3141  005226  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3142  005234  010412                          MOV     R4,(R2)         ;CLEAR SELECTED CAR
3143  005236  011203                          MOV     (R2),R3         ;GET THE WAS DATA
3144  005240  001741                          BEQ     1$              ;BR IF CAR GOT CLEARED
3145
3146  005242  004767  015424                  JSR     PC,SUER2        ;GO SET UP FOR ERROR CALL
3147  005246  012767  005224  173634          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
3148  005254  104007                          ERROR   7               ;FAILED TO CLR ALL BITS IN SELECTED CAR
3149  005256  000732                          BR      1$              ;GO TEST NEXT LINE
```

```
 3152                                   ;*****************************************************************
  (3)                                   ;*TEST 17       "BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
  (3)                                   ;*****************************************************************
  (2)  005260  000004                   TST17:  SCOPE
  (1)  005262  012767  000017  173744           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 3153                                   .REM    %
  (1)                                   TEST ABSTRACT:
  (1)                                   **************
  (1)
  (1)                                           THIS TEST VERIFIES THE ABILITY TO SET AND CLEAR ALL BITS IN
  (1)                                   ALL THE SELECTED LOCATIONS (LINES) OF THE BYTE COUNT MEMORY.
  (1)                                   IT USES THE CONFIGURATION PARAMETER (LINSEL:) TO DEFINE WHICH LINES TO
  (1)                                   TEST. THE TEST SEQUENCE IS AS FOLLOWS:
  (1)
  (1)                                           1. SELECT A LINE # TO TEST
  (1)                                           2. LOAD THE SELECTED LOCATION WITH 177777
  (1)                                           3. READ IT BACK TO VERIFY ALL BITS SET
  (1)                                           4. LOAD THE SELECTED LOCATION WITH 000000
  (1)                                           5. READ IT BACK TO VERIFY ALL BITS CLEARED
  (1)                                           6. REPEAT STEPS 1 THRU 5 UNTIL ALL SELECTED LINES ARE TESTED.
  (1)
  (1)                                   ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT
  (1)                                   LINE # IN SEQUENCE AS DEFINED BY "LINSEL".
  (1)
  (1)                                   ERRORS:
  (1)                                   ********
  (1)
  (1)                                   1.) [ERROR  10] IS CALLED TO REPORT ALL DATA COMPARE ERRORS
  (1)
  (1)                                   SYNC:
  (1)                                   *****
  (1)
  (1)                                   1.) WRITE 1'S:  M7277   SH4     LOAD SSR LOW BYTE H     CR1
  (1)
  (1)                                   2.) WRITE 0'S:  M7277   SH4     LOAD SSR HIGH BYTE H    CP2
  (1)
  (1)                                   DEBUG:  (REFER TO TEST 15)
  (1)                                   ******
  (1)
  (1)                                   KEY LOGIC:      (REFER TO TEST 15)
  (1)                                   **********
  (1)
  (1)                                   %
 3154  005270  010102                           MOV     R1,R2           ;COPY IT INTO R2
 3155  005272  062702  000010                   ADD     #BCR,R2         ;R2 GETS BCR ADDRESS
 3156  005276  004767  015522           1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
 3157  005302  000443                           BR      TST20           ;;BR IF DONE ALL SELECTED LINES
 3158  005304  012704  177777                   MOV     #-1,R4          ;RESULT IN BCR S/B = 177777
 3159  005310  156711  017630                   BISB    LINE,(R1)       ;SELECT A LINE NO.
 3160  005314  004567  015576                   JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
 3161  005320  025144                           LINE
 3162  005322  026046                           EM10+44
 3163
 3164  005324  112761  000000  000016   2$:     MOVB    #0,SSR(R1)      ;SCOPE SYNC
```

```
3165  005332  010412                          MOV     R4,(R2)         ;LOAD A BCR WITH ALL ONES
3166  005334  011203                          MOV     (R2),R3         ;GET THE WAS DATA FROM THE BCR
3167  005336  020403                          CMP     R4,R3           ;DID IT CONTAIN ALL ONES ??
3168  005340  001406                          BEQ     3$              ;;BR IF ALL 1'S
3169
3170  005342  004767  015324                  JSR     PC,SUER2        ;GO SET UP ERROR INFO
3171  005346  012767  005324  173534          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3172  005354  104010                          ERROR   10              ;FAILED TO SET ALL 1'S IN SELECTED BCR
3173
3174  005356  005004                  3$:     CLR     R4              ;RESULT IN BCR S/B = 000000
3175  005360  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3176  005366  010412                          MOV     R4,(R2)         ;CLEAR SELECTED BCR
3177  005370  011203                          MOV     (R2),R3         ;GET THE WAS DATA
3178  005372  001741                          BEQ     1$              ;BR IF BCR GOT CLEARED
3179
3180  005374  004767  015272                  JSR     PC,SUEP2        ;GO SET UP FOR ERROR CALL
3181  005400  012767  005356  173502          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
3182  005406  104010                          ERROR   10              ;FAILED TO CLR ALL BITS IN SELECTED BCR
3183  005410  000732                          BR      1$              ;GO TEST NEXT LINE
```

```
3186                                    ;*****************************************************************
 (3)                                    ;*TEST 20       "CAR" MEMORY PATTERNS TEST / 0'S DISTURB
 (3)                                    ;*****************************************************************
 (2)  005412  000004                    TST20:  SCOPE
 (1)  005414  012767  000020  173612            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3187                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT WHEN A TEST PATTERN IS WRITTEN INTO
 (1)                                    LOCATION "N" OF THE "CAR" MEMORY, IT DOES NOT DISTURB ANY BITS IN
 (1)                                    ANY OTHER LOCATIONS. THERE ARE THREE TEST PATTERNS USED* (177777,
 (1)                                    125252, 052525) FOR EACH LOCATION SELECTED BY THE CONFIGURATION PAR-
 (1)                                    AMETER "LINSEL". THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A TEST PATTERN
 (1)                                            2. SELECT A LINE # TO TEST
 (1)                                            3. CLEAR ALL 16. LOCATIONS IN THE MEMORY
 (1)                                            4. WRITE THE TEST PATTERN INTO THE SELECTED LOCATION
 (1)                                            5. VERIFY THAT THE PATTERN WAS WRITTEN CORRECTLY AND THAT
 (1)                                               NO OTHER LOCATIONS WERE DISTURBED.
 (1)                                            6. REPEAT 2 THRU 5 UNTIL ALL SELECTED LINES TESTED
 (1)                                            7. REPEAT 1 THRU 6 UNTIL ALL THREE PATTERNS TESTED
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH CHECKING THE
 (1)                                    NEXT LINE IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  46] IS CALLED TO REPORT ANY ERROR DETECTED. THE INFORM-
 (1)                                        ATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED,
 (1)                                        AND THE PATTERN USED.
 (1)
 (1)                                    SYNC*
 (1)                                    ******
 (1)
 (1)                                    1.) WRITE LINE: M7277   SHY     LOAD SSR LOW BYTE H     CR1
 (1)
 (1)                                    2.) READ CHECK: M7277   SHY     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:  (REFER TO TEST 14)
 (1)                                    ******
 (1)
 (1)                                    KEY LOGIC:      (REFER TO TEST 14)
 (1)                                    **********
 (1)                                    %
3188  005422  010102                            MOV     R1,R2           ;SET UP REGADR
3189  005424  062702  000006                    ADD     #CAR,R2
3190  005430  012705  024426                    MOV     #PATRNA,R5      ;SET UP POINTER TO DATA PATTERNS
3191  005434  012567  173542           1$:      MOV     (R5)+,$TMP1     ;GET A DATA TEST PATTERN
3192  005440  001472                            BEQ     TST21           ;;BR IF DONE THREE PATTERNS
3193  005442  004767  015356           11$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
3194  005446  000772                            BR      1$              ;BR IF DONE ALL SELECTED LINES
```

```
3195  005450  116757  017470  017470          MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOPING
3196
3197  005456  105067  173524          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
3198  005462  116711  173520          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
3199  005466  005012                          CLR     (R2)            ;CLR CAR FOR THAT LINE
3200  005470  105267  173512                  INCB    $TMP3           ;GENERATE NEW LINE NO.
3201  005474  126727  173506  000020          CMPB    $TMP3,#20       ;DONE CLEARING ALL LINES ?
3202  005502  001367                          BNE     3$              ;BR IF NOT
3203
3204  005504  116711  017436                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
3205  005510  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
3206  005516  016712  173460                  MOV     $TMP1,(R2)      ;LOAD CAR WITH TEST PATTERN
3207
3208  005522  105067  173456                  CLRB    $TMP2           ;INIT A LINE COUNTER
3209  005526  016704  173450          4$:     MOV     $TMP1,R4        ;SET UP S/B DATA
3210  005532  116711  173446                  MOVB    $TMP2,(R1)      ;SET LINE SELECT IN SCR
3211  005536  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3212  005544  011203                          MOV     (R2),R3         ;GET WAS DATA
3213  005546  126767  173432  017370          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
3214  00[illegible]  001401                   BEQ     5$              ;BR IF IT IS
3215  005556  005004                          CLR     R4              ;MAKE S/B DATA = 000000
3216  005560  020304                  5$:     CMP     R3,R4           ;CORRECT DATA IN CAR ?
3217  005562  001412                          BEQ     6$              ;BR IF YES
3218
3219  005564  004767  015170                  JSR     PC,SUER4        ;GO SET UP ERROR IN FO
3220  005570  004567  015322                  JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
3221  005574  [illegible]12[illegible]                  $TMP2
3222  005576  030620                          EM46+63
3223  005600  012767  005456  173302          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3224  005606  104046                          ERROR   46              ;INCORRECT DATA READ FROM CAR
3225
3226  005610  105267  173370          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
3227  005614  122767  000020  173362          CMPB    #20,$TMP2       ;DONE ALL LINES ?
3228  005622  001707                          BEQ     11$             ;BR IF YES
3229  005624  000740                          BR      4$              ;GO CHECK NEXT LINE
```

```
3232                                    ;*****************************************************************
 (3)                                    ;*TEST 21       "BCR" MEMORY PATTERNS TEST / 0'S DISTURB
 (3)                                    ;*****************************************************************
 (2)  005626  000004                    TST21:  SCOPE
 (1)  005630  012767  000021  173376            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3233                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT WHEN A TEST PATTERN IS WRITTEN INTO
 (1)                                    LOCATION "N" OF THE "BCR" MEMORY, IT DOES NOT DISTURB ANY BITS IN
 (1)                                    ANY OTHER LOCATIONS. THERE ARE THREE TEST PATTERNS USED# (177777,
 (1)                                    125252, 052525) FOR EACH LOCATION SELECTED BY THE CONFIGURATION PAR-
 (1)                                    AMETER "LINSEL". THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A TEST PATTERN
 (1)                                            2. SELECT A LINE # TO TEST
 (1)                                            3. CLEAR ALL 16. LOCATIONS IN THE MEMORY
 (1)                                            4. WRITE THE TEST PATTERN INTO THE SELECTED LOCATION
 (1)                                            5. VERIFY THAT THE PATTERN WAS WRITTEN CORRECTLY AND THAT
 (1)                                               NO OTHER LOCATIONS WERE DISTURBED.
 (1)                                            6. REPEAT 2 THRU 5 UNTIL ALL SELECTED LINES TESTED
 (1)                                            7. REPEAT 1 THRU 6 UNTIL ALL THREE PATTERNS TESTED
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH CHECKING THE
 (1)                                    NEXT LINE IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  47] IS CALLED TO REPORT ANY ERROR DETECTED. THE INFORM-
 (1)                                        ATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED,
 (1)                                        AND THE PATTERN USED.
 (1)
 (1)                                    SYNC#
 (1)                                    ******
 (1)
 (1)                                    1.) WRITE LINE: M7277   SHH     LOAD SSR LOW BYTE H     CR1
 (1)
 (1)                                    2.) READ CHECK: M7277   SHH     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:  (REFER TO TEST 15)
 (1)                                    ******
 (1)
 (1)                                    KEY LOGIC:      (REFER TO TEST 15)
 (1)                                    **********
 (1)                                    %
3234  005636  010102                            MOV     R1,R2           ;SET UP REGADR
3235  005640  062702  000010                    ADD     #BCR,R2
3236  005644  012705  024426                    MOV     #PATRNA,R5      ;SET UP POINTER TO DATA PATTERNS
3237  005650  012567  173326           1$:     MOV     (R5)+,$TMP1     ;GET 1 DATA TEST PATTERN
3238  005654  001472                            BEQ     TST22           ;;BR IF DONE THREE PATTERNS
3239  005656  004767  015142           11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
3240  005662  000772                            BR      1$              ;BR IF SELECTED ALL LINES
```

```
3241  005664  116767  017254  017254          MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOP
3242
3243  005672  105067  173310          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
3244  005676  116711  173304          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
3245  005702  005012                          CLR     (R2)            ;CLR BCR FOR THAT LINE
3246  005704  105267  173276                  INCB    $TMP3           ;GENERATE NEW LINE NO.
3247  005710  126727  173272  000020          CMPB    $TMP3,#20       ;DONE CLEARING ALL LINES ?
3248  005716  001367                          BNE     3$              ;BR IF NOT
3249
3250  005720  116711  017222                  MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
3251  005724  112761  000000  000016          MOVB    #0,SSR(R1)      ;SCOPE SYNC
3252  005732  016712  173244                  MOV     $TMP1,(R2)      ;LOAD BCR WITH TEST PATTERN
3253
3254  005736  105067  173242                  CLRB    $TMP2           ;INIT A LINE COUNTER
3255  005742  016704  173234          4$:     MOV     $TMP1,R4        ;SET UP S/B DATA
3256  005746  116711  173232                  MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
3257  005752  112761  000000  000017          MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3258  005760  011203                          MOV     (R2),R3         ;GET WAS DATA
3259  005762  126767  173216  017154          CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
3260  005770  001401                          BEQ     5$              ;BR IF IT IS
3261  005772  005004                          CLR     R4              ;MAKE S/B DATA = 000000
3262  005774  020304          5$:             CMP     R3,R4           ;CORRECT DATA IN BCR ?
3263  005776  001412                          BEQ     6$              ;BR IF YES
3264
3265  006000  004767  014754                  JSR     PC,SUER4        ;GO SET UP ERROR IN FO
3266  006004  004567  015106                  JSR     R5,SUNUM        ;GO SET UP LINE NO.IN MSG BUFFER
3267  006010  001204                          $TMP2
3268  006012  031020                          EM47+56
3269  006014  012767  005672  173066          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3270  006022  104047                          ERROR   47              ;INCORRECT DATA READ FROM BCR
3271
3272  006024  105267  173154          6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
3273  006030  122767  000020  173146          CMPB    #20,$TMP2       ;DONE ALL LINES ?
3274  006036  001707                          BEQ     11$             ;BR IF YES
3275  006040  000740                          BR      4$              ;GO CHECK NEXT LINE
```

```
3278                                    ;*****************************************************************
 (3)                                    ;*TEST 22       "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
 (3)                                    ;*****************************************************************
 (2)  006042  000004                    TST22:  SCOPE
 (1)  006044  012767  000022  173162            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3279                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT WHEN ALL ZEROS ARE WRITTEN INTO LINE "N"
 (1)                                    IN THE "CAR" MEMORY, IT DOES NOT CLEAR ANY BITS IN ANY OTHER LOCATIONS.
 (1)                                    ONLY THE LINES SELECTED BY "LINSEL" ARE TESTED. THE TEST SEQUENCE IS
 (1)                                    AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A LINE TO TEST
 (1)                                            2. SET ALL ONES (177777) INTO ALL MEMORY LOCATIONS
 (1)                                            3. CLEAR THE SELECTED LINE
 (1)                                            4. VERIFY THAT ONLY THE SELECTED LINE WAS CLEARED
 (1)                                               AND ALL OTHER LINES STILL CONTAIN 177777
 (1)                                            5. REPEAT STEPS 1 THRU 4 UNTIL ALL SELECTED LINES ARE TESTED
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT
 (1)                                    LINE IN SEQUENCE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  46] IS CALLED TO REPORT ALL ERRORS. THE INFORMATION PRINTED
 (1)                                        INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE
 (1)                                        PATTERN USED.
 (1)
 (1)                                    SYNC:
 (1)                                    *****
 (1)
 (1)                                    1.) WRITE LINE: M7277   SHH     LOAD SSR LOW BYTE H     CR1
 (1)
 (1)                                    2.) CHECK LINE: M7277   SHH     LOAD SSR HIGH BYTE H    CP2
 (1)
 (1)                                    DEBUG:  (REFER TO TEST 14)
 (1)                                    ******
 (1)
 (1)                                    KEY LOGIC:      (REFER TO TEST 14)
 (1)                                    **********
 (1)                                    %
3280  006052  010102                            MOV     R1,R2           ;SET UP REGADR
3281  006054  062702  000006                    ADD     #CAR,R2
3282  006060  012705  177777                    MOV     #-1,R5          ;TEST PATERRN IN R5 = 177777
3283  006064  010567  173112                    MOV     R5,$TMP1        ;SAVE FOR ERROR REPORTING
3284  006070  004767  014730            1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
3285  006074  000465                            BR      TST23           ;;BR IF DONE ALL LINES
3286  006076  116767  017042  017042            MOVB    LINE,LINEA      ;SAVE THE LINE NO. FOR ERROR LOOP
3287
3288  006104  105067  173076            2$:     CLRB    $TMP2           ;INIT LINE COUNTER
3289  006110  116711  173072            3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO CLEAR
```

```
3290  006114  010512                        MOV     R5,(R2)         ;LOAD CAR WITH 177777
3291  006116  105267  173064                INCB    $TMP3           ;GENERATE NEW LINE NO.
3292  006122  126727  173060  000020        CMPB    $TMP3,#20       ;DONE SETTING ALL LINES TO 177777 ?
3293  006130  001367                        BNE     3$              ;BR IF NOT
3294
3295  006132  116711  017010                MOVB    LINEA,(R1)      ;SET LINE SELECT IN SCR
3296  006136  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
3297  006144  005012                        CLR     (R2)            ;CLEAR THE CAR UNDER TEST
3298
3299  006146  105067  173032                CLRB    $TMP2           ;INIT A LINE COUNTER
3300  006152  005004                4$:     CLR     R4              ;MAKE S/B DATA = 000000
3301  006154  116711  173024                MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
3302  006160  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3303  006166  011203                        MOV     (R2),R3         ;GET WAS DATA
3304  006170  126767  173010  016746        CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
3305  006176  001401                        BEQ     5$              ;BR IF IT IS
3306  006200  010504                        MOV     R5,R4           ;MAKE S/B DATA = 177777
3307  006202  020304                5$:     CMP     R3,R4           ;CORRECT DATA IN CAR ?
3308  006204  001412                        BEQ     6$              ;BR IF YES
3309
3310  006206  004767  014546                JSR     PC,SUER4        ;GO SET UP ERROR IN F0
3311  006212  004567  014700                JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
3312  006216  001204                        $TMP2
3313  006220  030620                        EM46+63
3314  006222  012767  006104  172660        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3315  006230  104046                        ERROR   46              ;INCORRECT DATA READ FROM CAR
3316
3317  006232  105267  172746                INCB    $TMP2           ;GENERATE NEXT LINE NO.
3318  006236  122767  000020  172740        CMPB    #20,$TMP2       ;DONE ALL LINES ?
3319  006244  001711                        BEQ     1$              ;BR IF YES
3320  006246  000741                        BR      4$              ;GO CHECK NEXT LINE
```

```
                                ;*****************************************************************
                                ;*TEST 23       "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
                                ;*****************************************************************
006250  000004                  TST23:  SCOPE
006252  012767  000023  172754          MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    %
                                TEST ABSTRACT:
                                ***************

                                        THIS TEST VERIFIES THAT WHEN ALL ZEROS ARE WRITTEN INTO LINE "N"
                                IN THE "BCR" MEMORY, IT DOES NOT CLEAR ANY BITS IN ANY OTHER LOCATIONS.
                                ONLY THE LINES SELECTED BY "LINSEL" ARE TESTED. THE TEST SEQUENCE IS
                                AS FOLLOWS:

                                        1. SELECT A LINE TO TEST
                                        2. SET ALL ONES (177777) INTO ALL MEMORY LOCATIONS
                                        3. CLEAR THE SELECTED LINE
                                        4. VERIFY THAT ONLY THE SELECTED LINE WAS CLEARED
                                           AND ALL OTHER LINES STILL CONTAIN 177777
                                        5. REPEAT STEPS 1 THRU 4 UNTIL ALL SELECTED LINES ARE TESTED

                                ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT
                                LINE IN SEQUENCE.

                                ERRORS:
                                *******

                                1.) [ERROR  47] IS CALLED TO REPORT ALL ERRORS. THE INFORMATION PRINTED
                                    INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE
                                    PATTERN USED.

                                SYNC:
                                *****

                                1.) WRITE LINE: M7277   SHH     LOAD SSR LOW BYTE H     CR1

                                2.) CHECK LINE: M7277   SHH     LOAD SSR HIGH BYTE H    CP2

                                DEBUG:  (REFER TO TEST 15)
                                ******

                                KEY LOGIC:      (REFER TO TEST 15)
                                **********
                                %
006260  010102                          MOV     R1,R2           ;SET UP REGADR
006262  062702  000010                  ADD     #BCR,R2
006266  012705  177777                  MOV     #-1,R5          ;TEST PATERRN IN R5 = 177777
006272  010567  172704                  MOV     R5,$TMP1        ;SAVE IT FOR ERROR REPORTING
006276  004767  014522          1$:     JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
006302  000465                          BR      TST24           ;;BR IF DONE ALL LINES
006304  116767  016634  016634          MOVB    LINE,LINEA      ;SAVE THE LINE NO.

006312  105067  172670          2$:     CLRB    $TMP3           ;INIT LINE COUNTER
006316  116711  172664          3$:     MOVB    $TMP3,(R1)      ;SELECT A LINE TO INIT
```

```
3335  006322  010512                        MOV     R5,(R2)         ;LOAD BCR WITH 177777
3336  006324  105267  172656                INCB    $TMP3           ;GENERATE NEW LINE NO.
3337  006330  126727  172652  000020        CMPB    $TMP3,#20       ;DONE SETTING ALL LINES TO 177777 ?
3338  006336  001367                        BNE     3$              ;BR IF NOT
3339
3340  006340  116711  016602                MOVB    LINEA,(R1)      ;SET LINE SELECT BITS
3341  006344  112761  000000  000016        MOVB    #0,SSR(R1)      ;SCOPE SYNC
3342  006352  005012                        CLR     (R2)            ;CLEAR THE BCR UNDER TEST
3343
3344  006354  105067  172624                CLRB    $TMP2           ;INIT A LINE COUNTER
3345  006360  005004                4$:     CLR     R4              ;MAKE S/B DATA = 000000
3346  006362  116711  172616                MOVB    $TMP2,(R1)      ;SELECT A LINE TO CHECK
3347  006366  112761  000000  000017        MOVB    #0,SSR+1(R1)    ;SCOPE SYNC
3348  006374  011203                        MOV     (R2),R3         ;GET WAS DATA
3349  006376  126767  172602  016540        CMPB    $TMP2,LINE      ;IS THIS THE LINE WITH THE TEST PATTERN
3350  006404  001401                        BEQ     5$              ;BR IF IT IS
3351  006406  010504                        MOV     R5,R4           ;MAKE S/B DATA = 177777
3352  006410  020304                5$:     CMP     R3,R4           ;CORRECT DATA IN BCR ?
3353  006412  001412                        BEQ     6$              ;BR IF YES
3354
3355  006414  004767  014340                JSR     PC,SUER4        ;GO SET UP ERROR IN F0
3356  006420  004567  014472                JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG BUFFER
3357  006424  001204                        $TMP2
3358  006426  031020                        EM47+56
3359  006430  012767  006312  172452        MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3360  006436  104047                        ERROR   47              ;INCORRECT DATA READ FROM BCR
3361
3362  006440  105267  172540        6$:     INCB    $TMP2           ;GENERATE NEXT LINE NO.
3363  006444  122767  000020  172532        CMPB    #20,$TMP2       ;DONE ALL LINES ?
3364  006452  001711                        BEQ     1$              ;BR IF YES
3365  006454  000741                        BR      4$              ;GO CHECK NEXT LINE
```

```
3368                                    ;*****************************************************************
 (3)                                    ;#TEST 24       TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
 (3)                                    ;*****************************************************************
 (2)  006456  000004                    TST24:  SCOPE
 (1)  006460  012767  000024  172546            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3369                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT THE "EXT MEM" BITS (CAR<17:16> CAN
 (1)                                    BE SET AND CLEARED IN ALL "CAR" MEMORY LOCATIONS. IT WRITES THE
 (1)                                    BINARY TEST PATTERNS (11, 01, AND 10) INTO BITS<17:16> TO CHECK
 (1)                                    EVERY MEMORY LOCATION. THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SELECT A TEST PATTERN TO USE
 (1)                                            2. CLEAR ALL 18 BITS IN ALL 16 LOCATIONS
 (1)                                            3. SELECT A LINE TO TEST
 (1)                                            4. WRITE THE TEST PATTERN INTO <17:16> OF THE
 (1)                                               SELECTED LOCATION
 (1)                                            5. READ CHECK ALL LOCATIONS TO VERIFY THAT ONLY THE
 (1)                                               SELECTED LOCATION CONTAINS THE PATTERN
 (1)                                            6. REPEAT STEPS 3 THRU 5 UNTIL ALL SELECTED LINES TESTED
 (1)                                            7. REPEAT STEPS 1 THRU 6 UNTIL ALL PATTERNS USED
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT
 (1)                                    LINE IN SEQUENCE.
 (1)
 (1)                                            NOTES:  1.) BITS<05:04> IN THE "SCR" ARE USED TO WRITE THE
 (1)                                                       EXT MEM BITS
 (1)                                                    2.) BITS<07:06> IN THE "SSR" ARE USED TO CHECK BITS<17:16>
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  7] IS CALLED TO REPORT ALL ERRORS
 (1)
 (1)                                    SYNC:
 (1)                                    ******
 (1)
 (1)                                    1.) WRITE CAR:  M7277   SH4     LOAD LPR H      EP2
 (1)
 (1)                                    2.) READ CAR:   M7277   SH4     LOAD BCR H      FU2
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) ASSUMING THAT THE PREVIOUS "CAR" MEMORY TESTS RAN ERROR FREE, THE
 (1)                                        PROBLEM IS EITHER THE M7277 OR THE M7278
 (1)                                    2.) SET UP SCOPE ERROR LOOP AND START BACKTRACKING THROUGH THE LOGIC
 (1)                                        STARTING WITH THE KEY SIGNALS BELOW.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
```

```
(1)                                             M7277   SH5     SCR05 H         CD2
(1)                                                             SCR04 H         CE1
(1)                                                             SSR07 H         CF1
(1)                                                             SSR06 H         CH1
(1)
(1)                                             M7278   SH7     7415i MUX CHIPS E66 AND E58 (INPUT PIN 12)
(1)
(1)                                             NOTE:   THER MAY BE A PRINT ERROR ON SH7 OF THE M7278. THE
(1)                                                     SIGNALS INTO THE MUX CHIPS E66 AND E58 COME FROM THE
(1)                                                     M7277 SH5 RATHER FROM M7279 SH3.
(1)
(1)                                     %
3370  006466  010102                            MOV     R1,R2           ;SET UP REGADR
3371  006470  062702  000016                    ADD     #SSR,R2
3372  006474  012705  024436                    MOV     #^PATRN8,R5     ;SET UP POINTER TO DATA PATTERNS
3373  006500  012567  172476           1$:     MOV     (R5)+,$TMP1     ;GET THE PATTERNS
3374  006504  012567  172474                    MOV     (R5)+,$TMP2
3375  006510  001505                            BEQ     TST25           ;;BR IF DONE ALL PATTERNS
3376
3377  006512  105067  172470           2$:     CLRB    $TMP3           ;INIT A LINE COUNTER
3378  006516  142711  000017           3$:     BICB    #17,(R1)        ;INIT LINE SELECT BITS IN "SCR"
3379  006522  156711  172460                    BISB    $TMP3,(R1)      ;SELECT A LINE IN SCR
3380  006526  142711  000060                    BICB    #60,(R1)        ;SO WE CLEAR ALL THE MEM EXT BITS
3381  006532  005061  000006                    CLR     CAR(R1) ;CLEAR A CAR
3382  006536  105267  172444                    INCB    $TMP3           ;GENERATE NXT LINE NO.
3383  006542  122767  000020  172436           CMPB    #20,$TMP3       ;CLEARED THE WHOLE THING ?
3384  006550  001362                            BNE     3$              ;BR IF NOT
3385
3386  006552  004767  014246           4$:     JSR     PC,SELINE       ;GO SELECT A LINE NO.
3387  006556  000750                            BR      1$              ;BR IF DONE ALL LINES
3388  006560  156711  016360                    BISB    LINE,(R1)       ;SET UP LINE SELECT BITS
3389  006564  156711  172412                    BISB    $TMP1,(R1)      ;SET UP MEM EXT BIT PATTERN
3390  006570  012761  000000  000004            MOV     #0,LPR(R1)      ;SCOPE SYNC
3391  006576  012761  000000  000006            MOV     #0,CAR(R1)      ;WRITE EXT BITS IN THIS LOCATION
3392
3393  006604  105067  172400                    CLRB    $TMP4           ;INIT A LINE COUNTER
3394  006610  016704  172370           5$:     MOV     $TMP2,R4        ;SET UP S/B DATA
3395  006614  142711  000017                    BICB    #17,(R1)        ;INIT SELECT BITS IN "SCR"
3396  006620  156711  172364                    BISB    $TMP4,(R1)      ;SET SELECT BITS IN SCR
3397  006624  012761  000000  000014            MOV     #0,BKR(R1)      ;SCOPE SYNC
3398  006632  016103  000006                    MOV     CAR(R1),R3      ;READ THE SELECTED "CAR"
3399  006636  011203                            MOV     (R2),R3         ;GET THE WAS DATA
3400  006640  042703  177477                    BIC     #177477,R3      ;CLEAR JUNK BITS
3401  006644  126767  016274  172336           CMPB    LINE,$TMP4      ;LINE UNDER TEST ??
3402  006652  001401                            BEQ     6$              ;BR IF YES
3403  006654  005004                            CLR     R4              ;MAKE S/B DATA = 000000
3404  006656  020304                    6$:     CMP     R3,R4           ;WERE MEM EXT BITS CORRECT ?
3405  006660  001412                            BEQ     7$              ;BR IF YES
3406
3407  006662  004767  014004                    JSR     PC,SUER2        ;GO SET UP ERROR INFO
3408  006666  004567  014224                    JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG BUFFER
3409  006672  001210                            $TMP4
3410  006674  025777                            EM7+47
3411  006676  012767  006512  172204           MOV     #2$,$L.PERR     ;SET UP ERROR LOOP RETURN
```

```
3412  006704  104007                          ERROR   7               ;MEM EXT BITS READ INCORRECTLY
3413
3414  006706  105267  172276          7$:     INCB    $TMP4           ;GENERATE NXT LINE NO.
3415  006712  122767  000020  172270          CMPB    #20,$TMP4       ;DONE ALL LINES
3416  006720  001674                          BEQ     2$              ;BR IF YES
3417  006722  000732                          BR      5$              ;GO CHECK NEXT LINE
```

```
3420                                    ;*****************************************************************
 (3)                                    ;*TEST 25       TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED
 (3)                                    ;*****************************************************************
 (2)  006724  000004                    TST25:  SCOPE
 (1)  006726  012767  000025  172300            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3421                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT NO TRANSMITTER OR RECEIVER INTERRUPT
 (1)                                    OCCURS WHEN THE ENABLE BIT IS SET WITH OUT THE INTERRUPTING CONDITION
 (1)                                    ACTIVE. A BIT MASK (INTMSK:  030100) IS UESED TO DEFINE THE I.E. BITS.
 (1)                                    IN THE "SCR" (BITS 13, 12, AND 06). THE TEST SEQUENCE IS AS
 (1)                                    FOLLOWS:
 (1)
 (1)                                            1. SET UP THE XMIT AND RCVR VECTORS
 (1)                                            2. SELECT AN I.E. BIT TO TEST
 (1)                                            3. INIT THE SP AND LOCK OUT INTERRUPTS
 (1)                                            4. SET THE SELECTED BIT IN THE "SCR"
 (1)                                            5. CLEAR THE PSW TO ALLOW INTRS
 (1)                                            6. IF NO INTR:  REPEAT 2 THRU 5 UNTIL ALL BITS TESTED
 (1)                                            7. IF INTR:     REPORT ERROR AND CONTINUE WITH NEXT BIT TO TEST
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT BIT
 (1)                                    IN SEQUENCE .
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.) [ERROR  11] IS CALLED TO REPORT RCVR INTR FAULTS
 (1)
 (1)                                    2.) [ERROR  12] IS CALLED TO REPORT XMITTR INTR FAULTS
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.) PROBLEM IS MOST LIKELY THE M7289 MODULE IF THIS IS THE
 (1)                                        FIRST TEST TO FAIL.
 (1)                                    2.) SET UP SCOPE ERROR LOOP AND BACKTRACK THROUGH THE LOGIC
 (1)                                        STARTING WITH THE KEY LOGIC BELOW.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    ***********
 (1)
 (1)                                            M7289   SH6     XMIT INT REQ H  FM1
 (1)                                                            RCV INT REQ H   DP1
 (1)                                    %
3422  006734  012767  007012  172146            MOV     #3$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3423  006742  010102                            MOV     R1,R2           ;MAKE IT REGADR TOO
3424  006744  016703  015432                    MOV     DHVCT,R3        ;GET FIRST VECTOR ADDRESS
3425  006750  012723  007042                    MOV     #1$,(R3)+       ;GO TO 3$ IF RCVR INTRS
```

```
3426  006754  116723  016160          MOVB    DHRLVL,(R3)+
3427  006760  105723                  TSTB    (R3)+           ;UPDATE POINTER
3428  006762  012723  007064          MOV     #5$,(R3)+       ;GO TO 5$ IF XMITTR INTRS
3429  006766  116713  016147          MOVB    DHTLVL,(R3)
3430  006772  012705  000001          MOV     #1,R5           ;INIT BIT TEST MARKER
3431  006776  030567  015772    1$:   BIT     R5,INTMSK       ;TEST THIS BIT ??
3432  007002  001003                  BNE     3$              ;BR IF YES
3433  007004  006305            2$:   ASL     R5              ;SHIFT THE MARKER
3434  007006  001437                  BEQ     6$              ;BR IF TESTED ALL REQUIRED BITS
3435  007010  000772                  BR      1$              ;GO TEST FOR THIS ONE
3436
3437  007012  012706  001100    3$:   MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPING
3438  007016  004767  015276          JSR     PC,CHPS2        ;GO LOCK OUT INTRS
3439  007022  012711  004000          MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
3440  007026  010504                  MOV     R5,R4           ;SET UP S/B DATA
3441  007030  050511                  BIS     R5,(R1)         ;SET THE TEST I.E. BIT
3442  007032  004767  015246          JSR     PC,CHPS1        ;GO CLEAR PSW
3443  007036  000240                  NOP                     ;WAIT A BIT TO ALLOW INTR
3444  007040  000761                  BR      2$              ;OK - GO DO NEXT I.E. BIT
3445
3446  007042  004767  015266    4$:   JSR     PC,SAPS         ;SAVE THE ERROR PSW
3447  007046  011103                  MOV     (R1),R3         ;GET THE WAS DATA
3448  007050  004767  013622          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3449  007054  104011                  ERROR   11              ;DH11 RCVR SHOULD NOT HAVE INTERRUPTED
3450  007056  012716  007004          MOV     #2$,(SP)        ;SET UP TO RETURN
3451  007062  000002                  RTI                     ;RETURN TO TEST NEXT BIT
3452
3453  007064  004767  015244    5$:   JSR     PC,SAPS         ;SAVE THE ERROR PSW
3454  007070  011103                  MOV     (R1),R3         ;GET THE WAS DATA
3455  007072  004767  013600          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3456  007076  104012                  ERROR   12              ;XMITTER SHOULD NOT HAVE INTERRUPTED
3457  007100  012716  007004          MOV     #2$,(SP)        ;SET UP TO RETURN
3458  007104  000002                  RTI                     ;RETURN TO TEST NEXT BIT
3459
3460  007106  012706  001100    6$:   MOV     #STACK,SP       ;RESET THE SP JUST IN CASE
3461  007112  004767  015006          JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER IN VECTOR
```

```
3464                                    ;*****************************************************************
 (3)                                    ;*TEST 26       TEST CHAR. AVAIL  I.E.  WITH INTR. CONDITION ACTIVE
 (3)                                    ;*****************************************************************
 (2)  007116  000004                    TST26:  SCOPE
 (1)  007120  012767  000026  172106            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3465                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST USES MAINT. MODE (SCR09=1) TO SET THE CHAR AVAIL
 (1)                                    BIT (SCR7) TO GENERATE A RCVR INTERRUPT THROUGH THE PROPER VECTOR.
 (1)                                    THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SET UP THE VECTORS
 (1)                                            2. ISSUE DH11 "MASTER CLR", RESET THE SP, AND LOCK OUT INTRS
 (1)                                            3. PRIMT THE DH11 TO GENERATE A RCVR INTR
 (1)                                            4. CLEAR THE PSW TO ALLOW INTRS
 (1)                                            5. REPORT NO RCVR INTR OR FALSE XMITTR INTR.
 (1)
 (1)                                    ALL ERRORS ARE REPORTED AND THEN THE TEST RESETS THE VECTOR AND SP
 (1)                                    AND CONTINUES TO THE NEXT TEST IN THE PROGRAM.
 (1)
 (1)                                    ERRORS:
 (1)                                    ********
 (1)
 (1)                                    1.) [ERROR  13] IS CALLED TO REPORT RCVR INTR FAULTS
 (1)                                    2.) [ERROR  12] IS CALLED TO REPORT XMITTR INTR FAULTS
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    *******
 (1)
 (1)                                    1.) IF NO RCVR INTR OCCURRED THE PROBLEM IS EITHER SECTION "A" OF THE
 (1)                                        M7821 OR THE M7289 - SH6.
 (1)                                    2.) IF A FALSE XMIT INTR OCCURRED THE PROBLEM IS MOST LIKELY THE
 (1)                                        M7821 GENERATING AN INCORRECT VECTOR ADDRESS.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7289   SH6     E31-12
 (1)
 (1)                                            M7821   SEC "A" BUS A BR L      U2
 (1)                                                            BUS SACK L      T2
 (1)                                                            BUS BG IN H     B1
 (1)                                                            "A" MASTER L    N1
 (1)                                                            VECTOR BIT 02 H D2
 (1)
 (1)                                            NOTE:   REMEMBER THAT PROBLEMS IN THIS AREA COULD BE CAUSED
 (1)                                                    BY ANY DEVICE IN THE SYSTEM INCLUDING THE "CPU".
 (1)                                                    SYSTEM RE-CONFIGURATION TO ISOLATE THE FAULTY SUB-
 (1)                                                    SYSTEM MAY BE REQUIRED.
```

```
(1)                                     %
3466  007126  012767  007164  171754          MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3467  007134  010102                          MOV     R1,R2           ;MAKE IT REGADR TOO
3468  007136  016703  015240                  MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
3469  007142  012723  007274                  MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
3470  007146  116723  015766                  MOVB    DHRLVL,(R3)+
3471  007152  105723                          TSTB    (R3)+           ;UPDATE POINTER
3472  007154  012723  007250                  MOV     #2$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
3473  007160  116713  015755                  MOVB    DHTLVL,(R3)
3474  007164  012711  004000          1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
3475  007170  012706  001100                  MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
3476  007174  004767  015120                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
3477  007200  012711  001000                  MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
3478  007204  052711  000100                  BIS     #BIT06,(R1)     ;SET CHAR AVAILABLE I.E. BIT
3479  007210  052711  000200                  BIS     #BIT07,(R1)     ;SET THE CHAR AVAIL BIT TO FORCE INTR
3480  007214  004767  015064                  JSR     PC,CHPS1        ;GO CLEAR PSW
3481  007220  000240                          NOP                     ;GIVE IT A LITTLE TIME
3482
3483  007222  004767  015106                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
3484  007226  011103                          MOV     (R1),R3         ;GET THE WAS DATA
3485  007230  005011                          CLR     (R1)            ;CLEAR OUT THE SCR
3486  007232  005011                          CLR     (R1)
3487  007234  012704  001300                  MOV     #1300,R4        ;SET UP S/B DATA
3488  007240  004767  013432                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3489  007244  104013                          ERROR   13              ;TIMED OUT AWAITING CHAR AVAIL INTR
3490  007246  000412                          BR      3$              ;GO EXIT TEST
3491
3492  007250  004767  015060          2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
3493  007254  011103                          MOV     (R1),R3         ;GET WAS DATA
3494  007256  012704  001300                  MOV     #1300,R4        ;SET UP S/B DATA
3495  007262  005011                          CLR     (R1)            ;CLR OUT SCR REG
3496  007264  005011                          CLR     (R1)
3497  007266  004767  013404                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3498  007272  104012                          ERROR   12              ;UNEXPECTED XMITTR INTR
3499
3500  007274  012706  001100          3$:     MOV     #STACK,SP       ;RESET THE SP
3501  007300  004767  014620                  JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
3504                                    ;*****************************************************************
 (3)                                    ;*TEST 27       TEST SILO OVFLW.  I.E.  WITH INTR. CONDITION ACTIVE
 (3)                                    ;*****************************************************************
 (2)  007304  000004                    TST27:  SCOPE
 (1)  007306  012767  000027  171720            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3505                                    .REM    X
 (1)                                    TEST ABSTRACT
 (1)                                    *************
 (1)
 (1)                                            THIS TEST USES MAINT. MODE (SCR09=1) TO ENABLE SILO FULL
 (1)                                    INTERRUPT  THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1.  SET UP XMIT AND RCVR VECTORS
 (1)                                            2.  RESET THE DH11 AND S.P. - THE LOCK OUT INTRS.
 (1)                                            3.  PRIME DH11 TO GENERATE SILO FULL INTR. - ALLOW INTRS.
 (1)                                            4.  REPORT ERROR IF NO RCVR. INTR OCCURS OR
 (1)                                                A FALSE XMIT INTR. DOES OCCUR
 (1)                                            5.  AFTER REPORTING ANY ERRORS DETECTED RESET
 (1)                                                THE SP AND VECTORS THEN GO TO TEST 30
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 43] IS CALLED TO REPORT NO RCVR INTR OCCURRED
 (1)                                    2.  [ERROR 12] IS CALLED TO REPORT FALSE TRANSMITTER INTR.
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  IF THE RECEIVER INTR FAILED TO INTERRUPT PROBLEM IS MOST LIKELY
 (1)                                        THE M7289 MODULE
 (1)                                    2.  IF A FALSE XMITTR INTR OCCURRED PROBLEM IS MOST LIKELY THE M7281 MODULE
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7289   SHG     E3S,ES0, OR E31
 (1)                                                            SCR 14 H (STORAGE) DS1
 (1)
 (1)                                            M7821   "B" SECTION
 (1)
 (1)                                    %
3506  007314  012767  007352  171566            MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3507  007322  010102                            MOV     R1,R2           ;MAKE IT REGADR TOO
3508  007324  016703  015052                    MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
3509  007330  012723  007462                    MOV     #3$,(R3)+       ;GO TO 3$ IF RCVR INTRS
3510  007334  116723  015600                    MOVB    DHRLVL,(R3)+
3511  007340  105723                            TSTB    (R3)+           ;UPDATE POINTER
3512  007342  012723  007436                    MOV     #2$,(R3)+       ;GO TO 2$ ON XMITTR INTRS
3513  007346  116713  015567                    MOVB    DHTLVL,(R3)
3514  007352  012711  004000            1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
```

```
3515  007356  012706  001100              MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
3516  007362  004767  014732              JSR     PC,CHPS2        ;GO LOCK OUT INTRS
3517  007366  012711  001000              MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
3518  007372  052711  040000              BIS     #BIT14,(R1)     ;SET SILO OVFLW I.E. BIT
3519  007376  052711  010000              BIS     #BIT12,(R1)     ;SET THE SILO FULL BIT TO FORCE INTR
3520  007402  004767  014676              JSR     PC,CHPS1        ;GO CLEAR PSW
3521  007406  000240                      NOP                     ;GIVE IT A LITTLE TIME
3522
3523  007410  004767  014720              JSR     PC,SAPS         ;SAVE THE ERROR PSW
3524  007414  011103                      MOV     (R1),R3         ;GET THE WAS DATA
3525  007416  005011                      CLR     (R1)            ;CLEAR OUT THE SCR
3526  007420  005011                      CLR     (R1)
3527  007422  012704  051000              MOV     #51000,R4       ;SET UP S/B DATA
3528  007426  004767  013244              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3529  007432  104043                      ERROR   43              ;TIMED OUT AWAITING SILO OVFLW INTR
3530  007434  000412                      BR      3$              ;GO EXIT TEST
3531
3532  007436  004767  014672      2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
3533  007442  011103                      MOV     (R1),R3         ;GET WAS DATA
3534  007444  012704  051000              MOV     #51000,R4       ;SET UP S/B DATA
3535  007450  005011                      CLR     (R1)            ;CLR OUT SCR REG
3536  007452  005011                      CLR     (R1)
3537  007454  004767  013216              JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3538  007460  104012                      ERROR   12              ;UNEXPECTED XMITTR INTR
3539
3540  007462  012706  001100      3$:     MOV     #STACK,SP       ;RESET THE SP
3541  007466  004767  014432              JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
3544                                    ;*****************************************************************
 (3)                                    ;*TEST 30       TEST NON EX MEM I.E.  WITH INTR. CONDITION ACTIVE
 (3)                                    ;*****************************************************************
 (2)  007472  000004                    TST30:  SCOPE
 (1)  007474  012767  000030  171532            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3545                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT THE NON-EX-MEM BIT (SCR10) CAN CAUSE A
 (1)                                    TRANSMITTER INTERRUPT VIA THE PROPER VECTOR. THE TEST SEQUENCE IS AS
 (1)                                    FOLLOWS:
 (1)
 (1)                                            1.  SET UP XMIT AND RCVR VECTORS
 (1)                                            2.  CLEAR THE DH11, RESET SP, AND LOCK OUT INTRS
 (1)                                            3.  PRIME DH11 TO GENERATE XMIT INTR IN MAINT. MODE
 (1)                                            4.  ALLOW INTRS.
 (1)                                            5.  REPORT ERROR IF NO XMIT INTR OCCURS OR IF
 (1)                                                A FALSE RCVR INTR OCCURS
 (1)                                            6.  REST SP AND VECTORS THEN GO TO TEST 31
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 44] IS CALLED IF NON-EX-MEM FAILS TO GENERATE XMIT INTR
 (1)                                    2.  [ERROR 11] IS CALLED IF FALSE RCVR INTR OCCURS
 (1)
 (1)                                    SYNC:   M7277   SW3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  IF THE NON-EX-MEM INTERRUPT FAILS TO OCCUR PROBLEM IS MOST LIKELY
 (1)                                        THE M7289 MODULE
 (1)                                    2.  IF A FALSE RCVR INTR OCCURS PROBLEM IS MOST LIKELY THE M7289 OR
 (1)                                        THE M7281 MODULES.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7289   SW6     SCR 10 H (NO EX MEM)     FL1
 (1)                                                            E35, E41, OR E48
 (1)
 (1)                                    %
3546  007502  012767  007540  171400            MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3547  007510  010102                            MOV     R1,R2           ;MAKE IT REGADR TOO
3548  007512  016703  014664                    MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
3549  007516  012723  007624                    MOV     #2$,(R3)+       ;GO TO 2$ IF RCVR INTRS
3550  007522  116723  015412                    MOVB    DHRLVL,(R3)+
3551  007526  105723                            TSTB    (R3)+           ;UPDATE POINTER
3552  007530  012723  007650                    MOV     #3$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
3553  007534  116713  015401                    MOVB    DHTLVL,(R3)
3554  007540  012711  004000            1$:     MOV     #BIT11,(R1)     ;CLR THE DH11
```

```
3555  007544  012706  001100            MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
3556  007550  004767  014544            JSR     PC,CHPS2        ;GO LOCK OUT INTRS
3557  007554  012711  001000            MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
3558  007560  052711  020000            BIS     #BIT13,(R1)     ;SET XMITTR I.E. BIT
3559  007564  052711  002000            BIS     #BIT10,(R1)     ;SET THE NON EX MEM BIT TO FORCE INTR
3560  007570  004767  014510            JSR     PC,CHPS1        ;GO CLEAR PSW
3561  007574  000240                    NOP                     ;GIVE IT A LITTLE TIME
3562
3563  007576  004767  014532            JSR     PC,SAPS         ;SAVE THE ERROR PSW
3564  007602  011103                    MOV     (R1),R3         ;GET THE WAS DATA
3565  007604  005011                    CLR     (R1)            ;CLEAR OUT THE SCR
3566  007606  005011                    CLR     (R1)
3567  007610  012704  023000            MOV     #23000,R4       ;SET UP S/B DATA
3568  007614  004767  013056            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3569  007620  104044                    ERROR   44              ;TIMED OUT AWAITING NON EX MEM INTR
3570  007622  000412                    BR      3$              ;GO EXIT TEST
3571
3572  007624  004767  014504    2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
3573  007630  011103                    MOV     (R1),R3         ;GET WAS DATA
3574  007632  012704  023000            MOV     #23000,R4       ;SET UP S/B DATA
3575  007636  005011                    CLR     (R1)            ;CLR OUT SCR REG
3576  007640  005011                    CLR     (R1)
3577  007642  004767  013030            JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3578  007646  104011                    ERROR   11              ;UNEXPECTED RCVR INTR
3579
3580  007650  012706  001100    3$:     MOV     #STACK,SP       ;RESET THE SP
3581  007654  004767  014244            JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
3584                                    ;*****************************************************************
 (3)                                    ;*TEST 31       TEST XMITTR DONE I.E.  WITH INTR. CONDITION ACTIVE
 (3)                                    ;*****************************************************************
 (2)  007660  000004                    TST31:  SCOPE
 (1)  007662  012767  000031  171344            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3585                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT XMIT DONE (SCR15) CAN BE SET IN MAINT.
 (1)                                    MODE TO CAUSE A XMITTR INTR VIA THE PROPER VECTOR. THE TEST SEQUENCE
 (1)                                    IS AS FOLLOWS:
 (1)
 (1)                                            1.  SET UP XMIT AND RCVR VECTORS
 (1)                                            2.  CLEAR THE DH11, RESET SP, AND LOCK OUT INTRS
 (1)                                            3.  PRIME DH11 TO GENERATE "XMIT DONE" INTR
 (1)                                            4.  CLEAR PSW TO ALLOW INTRS
 (1)                                            5.  REPOT ERROR IF XMITTR FAILS TO INTR OR A
 (1)                                                FALSE RCVR INTR OCCURS
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 45] IS CALLED TO REPORT "XMIT DONE" INTR FAILED TO OCCUR
 (1)                                    2.  [ERROR 111] IS CALLED TO REPORT FALSE RCVR INTRS
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  IF NO XMIT INTR OCCURS PROBLEM IS MOST LIKELY THE M7289 MODULE
 (1)                                    2.  IF A FALSE RCVR INTR. OCCURS PROBLEM IS MOST LIKELY THE M7821 MODULE.
 (1)
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7289   SH6     SCR 15 H (XMIT) FR2
 (1)                                                            E48, E50
 (1)
 (1)                                    %
3586  007670  012767  007726  171212            MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3587  007676  010102                            MOV     R1,R2           ;MAKE IT REGADR TOO
3588  007700  016703  014476                    MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
3589  007704  012723  010012                    MOV     #2$,(R3)+       ;GO TO 2$ IF RCVR INTRS
3590  007710  116723  015224                    MOVB    DHRLVL,(R3)+
3591  007714  105723                            TSTB    (R3)+           ;UPDATE POINTER
3592  007716  012723  010036                    MOV     #3$,(R3)+       ;GO TO 3$ ON XMITTR INTRS
3593  007722  116713  015213                    MOVB    DHTLVL,(R3)
3594  007726  012711  004000          1$:       MOV     #BIT11,(R1)     ;CLR THE DH11
3595  007732  012706  001100                    MOV     #STACK,SP       ;RESET THE SP FOR ERROR LOOPS
3596  007736  004767  014356                    JSR     PC,CHPS2        ;GO LOCK OUT INTRS
```

```
3597  007742  012711  001000                MOV     #BIT09,(R1)     ;SET MAINT MODE BIT
3598  007746  052711  020000                BIS     #BIT13,(R1)     ;SET XMIT DONE I.E. BIT
3599  007752  052711  100000                BIS     #BIT15,(R1)     ;SET THE XMITTR DONE BIT TO FORCE INTR
3600  007756  004767  014322                JSR     PC,CHPS1        ;GO CLEAR PSW
3601  007762  000240                        NOP                     ;GIVE IT A LITTLE TIME
3602
3603  007764  004767  014344                JSR     PC,SAPS         ;SAVE THE ERROR PSW
3604  007770  011103                        MOV     (R1),R3         ;GET THE WAS DATA
3605  007772  005011                        CLR     (R1)            ;CLEAR OUT THE SCR
3606  007774  005011                        CLR     (R1)
3607  007776  012704  121000                MOV     #121000,R4      ;SET UP S/B DATA
3608  010002  004767  012670                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3609  010006  104045                        ERROR   45              ;TIMED OUT AWAITING XMIT DONE INTR
3610  010010  000412                        BR      3$              ;GO EXIT TEST
3611
3612  010012  004767  014316        2$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
3613  010016  011103                        MOV     (R1),R3         ;GET WAS DATA
3614  010020  012704  121000                MOV     #121000,R4      ;SET UP S/B DATA
3615  010024  005011                        CLR     (R1)            ;CLR OUT SCR REG
3616  010026  005011                        CLR     (R1)
3617  010030  004767  012642                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3618  010034  104011                        ERROR   11              ;UNEXPECTED RCVR INTR
3619
3620  010036  012706  001100        3$:     MOV     #STACK,SP       ;RESET THE SP
3621  010042  004767  014056                JSR     PC,RESTRP       ;GO RESTORE TRAP CATCHER
```

```
3624                                    ;*****************************************************************
 (3)                                    ;*TEST 32       BASIC TRANSMITTER "NPR" LOGIC TEST 1
 (3)                                    ;*****************************************************************
 (2)  010046  000004                    TST32:  SCOPE
 (1)  010050  012767  000032  17115     .       MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3625                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST TRANSMITS A SINGLE BYTE FROM LOCATION 0 ON ALL SELECTED
 (1)                                    LINES (AS SELECTED BY THE CONFIGURATION PARAMETER "LINSEL:") ONE AT A TIME.
 (1)                                    THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1.  SET UP THE XMITTR VECTOR
 (1)                                            2.  SELECT A LINE # TO TEST
 (1)                                            3.  RESET THE SP
 (1)                                            4.  CLEAR ALL LOCATIONS IN "CAR" AND "BCR" MEMORIES
 (1)                                            5.  LOCK OUT INTRS AND CLEAR THE DH11
 (1)                                            6.  PRIME DH11 TO XMIT ONE CHAR FROM LOCATION 0
 (1)                                                [9600 BAUD, 5-BITS, 1 STOP BIT]
 (1)                                            7.  ACTIVATE SELECTED XMITTR AND ENABLE XMIT DONE INTR
 (1)                                            8.  CLEAR PSW TO ALLOW INTR.
 (1)                                            9.  ACTIVATE TIMER TO WAIT FOR XMIT DONE INTR
 (1)                                            10. IF TIMEOUT OCCURS REPORT ERROR AND RESTART AT STEP 2
 (1)                                            11. IF XMIT INTERRUPT OCCURS CHECK THE FOLLOWING CONDITIONS
 (1)                                                AND REPORT ANY ERRORS:
 (1)
 (1)                                                A.  XMIT DONE (SCR15=1) SET
 (1)                                                B.  "BAR" BIT GOT CLEARED
 (1)                                                C.  "CAR" REGISTER GOT INCREMENTED TO +1
 (1)                                                D.  "BCR" REGISTER GO INCREMENTED TO 0
 (1)
 (1)                                            12. REPEAT STEP 2 THRU 11 UNTIL ALL SELECTED LINES TESTED
 (1)                                            13. AFTER TESTING ALL LINES CLEAR THE DH11, RESET THE VECTOR
 (1)                                                CLEAR PSW, RESET SP, AND GO TO TEST 33.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 15] IS CALLED IF XMIT DONE FAILS TO INTR ON TIME
 (1)                                    2.  [ERROR 14] IS CALLED IF XMIT DONE NOT SET
 (1)                                    3.  [ERROR 14] IS CALLED IF "BAR" BIT FAILED TO CLEAR
 (1)                                    4.  [ERROR 14] IS CALLED IF "CAR" NOT INCREMENTED PROPERLY
 (1)                                    5.  [ERROR 14] IS CALLED IF "BCR" NOT INCREMENTED PROPERLY
 (1)
 (1)                                        ALL ERROR MESSAGE HEADERS INCLUDE THE LINE NO. OF THE FAILING
 (1)                                        LINE.
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)                                            (NOTE: USE SR09=1 TO LOCK ON FAILING LINE AND SR13=1 TO
 (1)                                             INHIBIT ERROR PRINTOUT TO MINIMIZE SCOPE LOOP.)
 (1)
 (1)                                    DEBUG:
```

```
(1)                                     ******
(1)
(1)                                     1.  IF ALL LINES FAIL TO INTERRUPT ON TIME, SUSPECT LOSS OF 9600 BAUD
(1)                                         CLOCK SIGNAL OR BYTE COUNT DECODER ON M7278 SH3 (XMIT FINISHED PULSE L)
(1)                                     2.  IF GROUP OF 8 LINES <15:08> OR <07:00> FAIL TO INTERRUPT ON TIME, SUSPECT
(1)                                         THE BUFFERED CLOCK SIGNALS (TOP AND BOT) ON THE M7288 SH3
(1)                                     3.  IF ONLY ONE LINE FAILS TO INTERRUPT ON TIME, SUSPECT LOSS OF CLOCK IN
(1)                                         LINE MULTIPLEXORS M7288 SH4-SH11.
(1)                                     4.  IF LINE INTERRUPTED OK BUT "SCR", "BAR", "CAR", OR "BCR" WAS INCORRECT
(1)                                         REFER TO KEY LOGIC SIGNALS BELOW.
(1)
(1)
(1)                                     KEY LOGIC:
(1)                                     **********
(1)
(1)                                     XMIT DONE FAILED TO SET:
(1)
(1)                                             M7289   SH6     SCR15 H FR2
(1)                                             M7278   SH3     XMIT FINISHED PULSE L   AR1
(1)
(1)                                     "BAR" BIT FAILED TO CLEAR:
(1)
(1)                                             M7278   SH3     CLR BAR <15:00> L
(1)                                                     SH5     BAR <15:12> H   E54, E55 (7474)
(1)                                                     SH6     BAR <11:08> H   E62, E63 (7474)
(1)                                                     SH7     BAR <07:04> H   E70, E71 (7474)
(1)                                                     SH8     BAR <03:00> H   E78, E79 (7474)
(1)
(1)                                     "CAR" REG NOT INCREMENTED:
(1)
(1)                                             M7277   SH6     END CYCLE PULSE DLY H   FL2
(1)                                                     SH5 AND SH6 "CAR" MEMORY LOGIC
(1)                                             M796            END CYCLE H P2
(1)
(1)                                     "BCR" REG NOT INCREMENTED:
(1)
(1)                                             M796    END CYCLE L     N2
(1)                                             M7278   SH3 AND SH4 "BCR" MEMORY LOGIC
(1)
(1)                                     %
3626  010056  012767  010112  171024          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3627  010064  016703  014312                  MOV     DHVCT,R3        ;GET THE FIRST VECTOR ADDRESS
3628  010070  062703  000004                  ADD     #4,R3           ;POINT TO XMITTR ENTRY
3629  010074  012723  010256                  MOV     #4$,(R3)+       ;GO TO 4$ ON XMITTR INTR
3630  010100  116713  015035                  MOVB    DHTLVL,(R3)
3631  010104  004767  012714          1$:     JSR     PC,SELINE       ;GO SELECT A LINE NO. TO TEST
3632  010110  000552                          BR      8$              ;BR IF TESTED ALL SELECTED LINES
3633
3634  010112  012706  001100          2$:     MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
3635  010116  010102                          MOV     R1,R2           ;SET UP REGADR
3636  010120  012704  120000                  MOV     #120000,R4      ;SET UP S/B DATA
3637  010124  156704  015014                  BISB    LINE,R4
3638  010130  004767  013042                  JSR     PC,CLCABC       ;GO CLEAR CAR AND BCR MEMORIES
3639  010134  004767  014160                  JSR     PC,CHPS2        ;GO LOCK OUT INTRS
```

```
3640  010140  012711  004000                    MOV     #BIT11,(R1)     ;CLEAR THE DH11 INTERFACE
3641  010144  156711  014774                    BISB    LINE,(R1)       ;SELECT A LINE NO.
3642  010150  012761  177777  000010            MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
3643  010156  012761  033500  000004            MOV     #33500,LPR(R1)  ;SET UP LINE PARAMETERS
3644  010164  056761  014222  000012            BIS     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
3645  010172  052711  020000                    BIS     #BIT13,(R1)     ;ENABLE INTERRUPT ON XMIT DONE
3646  010176  004767  014102                    JSR     PC,CHPS1        ;GO CLEAR PSW
3647
3648  010202  012767  000001  014762            MOV     #1,TIMEA        ;INIT TIMER A
3649  010210  005067  014760                    CLR     TIMEB           ;INIT TIMER B
3650  010214  000240                    3$:     NOP                     ;DO NOTHING WAIT
3651  010216  004767  013730                    JSR     PC,TIMEIT       ;CALL TIMER
3652  010222  000774                    BR      3$                      ;TIMER ROUTINE WILL MOVE RETURN PC AROUND
3653                                            ;THIS BRANCH IF TIMEOUT OCCURS
3654
3655  010224  004767  014104                    JSR     PC,SAPS         ;SAVE THE ERROR PSW
3656  010230  011103                    MOV     (R1),R3                 ;GET THE WAS DATA
3657  010232  042703  000200                    BIC     #BIT07,R3       ;WE'RE NOT INTERESTED IN THIS BIT
3658  010236  004767  012434                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3659  010242  004567  012650                    JSR     R5,SUNUM        ;GO SET LINE NO. IN ERROR MSG
3660  010246  025144                    LINE
3661  010250  026344                    EMIS+43
3662  010252  104015                    ERROR   15                      ;TIMEOUT WHILE AWAITING XMIT INTR
3663  010254  000713                    BR      1$                      ;GO TEST NEXT LINE
3664
3665  010256  005711                    4$:     TST     (R1)            ;DID XMIT DONE SET ??
3666  010260  100411                    BMI     5$                      ;BR IF YES
3667
3668  010262  004767  014046                    JSR     PC,SAPS         ;SAVE THE ERROR PSW
3669  010266  011103                    MOV     (R1),R3                 ;GET THE WAS DATA
3670  010270  004767  012402                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3671  010274  004767  000174                    JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3672  010300  104014                    ERROR   14                      ;XMIT DONE FAILED TO SET
3673  010302  000700                    BR      1$                      ;GO TEST NEXT LINE
3674
3675  010304  016103  000012            5$:     MOV     BAR(R1),R3      ;GET WAS DATA FROM "BAR"
3676  010310  001413                    BEQ     6$                      ;BR IF BAR BIT GOT CLEARED
3677
3678  010312  004767  014016                    JSR     PC,SAPS         ;SAVE THE ERROR PSW
3679  010316  062702  000012                    ADD     #BAR,R2         ;SET UP REGADR
3680  010322  005004                    CLR     R4                      ;SET UP S/B DATA
3681  010324  004767  012346                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3682  010330  004767  000140                    JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3683  010334  104014                    ERROR   14                      ;BAR BIT FAILED TO CLEAR
3684  010336  000662                    BR      1$                      ;GO TEST NEXT LINE
3685
3686  010340  016103  000006            6$:     MOV     CAR(R1),R3      ;GET THE WAS DATA FROM CAR
3687  010344  022703  000001                    CMP     #1,R3           ;DID IT GET INCREMENTED ?
3688  010350  001414                    BEQ     7$                      ;BR IF YES
3689
3690  010352  004767  013756                    JSR     PC,SAPS         ;SAVE THE ERROR PSW
3691  010356  012704  000001                    MOV     #1,R4           ;SET UP S/B DATA
3692  010362  062702  000006                    ADD     #CAR,R2         ;SET UP REGADR
3693  010366  004767  012304                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
```

```
3694  010372  004767  000076          JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3695  010376  104014                  ERROR   14              ;CAR REG NOT  INCREMENTED PROPERLY
3696  010400  000641                  BR      1$              ;GO TEST NEXT LINE
3697
3698  010402  016103  000010    7$:   MOV     BCR(R1),R3      ;GET WAS DATA FROM BCR
3699  010406  001636                  BEQ     1$              ;BR IF BCR GOT INCREMENTED TO 000000
3700
3701  010410  004767  013720          JSR     PC,SAPS         ;SAVE THE ERROR PSW
3702  010414  005004                  CLR     R4              ;SET UP S/B DATA
3703  010416  062702  000010          ADD     #BCR,R2         ;SET UP REGADR
3704  010422  004767  012250          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3705  010426  004767  000042          JSR     PC,9$           ;GO SET UP SOME ERROR STUFF
3706  010432  104014                  ERROR   14              ;BCR REG NOT INCREMENTED PROPERLY
3707  010434  000623                  BR      1$              ;GO TEST NEXT LINE
3708
3709  010436  012711  004000    8$:   MOV     #BIT11,(R1)     ;CLEAR THE DH11
3710  010442  016703  013734          MOV     DHVCT,R3        ;GET THE VECTOR ADDR
3711  010446  062703  000004          ADD     #4,R3           ;POINT TO XMIT VECTOR
3712  010452  010313                  MOV     R3,(R3)         ;RESTORE TRAP CATCHER
3713  010454  062723  000002          ADD     #2,(R3)+
3714  010460  005013                  CLR     (R3)
3715  010462  004767  013616          JSR     PC,CHPS1        ;GO CLEAR PSW
3716  010466  012706  001100          MOV     #STACK,SP       ;RESET THE STACK POINTER
3717  010472  000405                  BR      TST33           ;;GO TO NEXT TEST
3718
3719  010474  004567  012416    9$:   JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MSG.
3720  010500  025144                  LINE
3721  010502  026276                  EM14+44
3722  010504  000207                  RTS     PC              ;RETURN TO REPORT ERROR
```

```
3725                                    ;*****************************************************************
 (3)                                    ;*TEST 33       TRANSMITTR NPR LOGIC TEST 2
 (3)                                    ;*****************************************************************
 (2)  010506  000004                    TST33:  SCOPE
 (1)  010510  012767  000033  170516            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3726                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST IS SIMILAR TO TEST 32 EXCEPT THAT ALL LOCATIONS IN THE "BCR"
 (1)                                    AND "CAR" MEMORIES ARE TESTED TO VERIFY THAT TRANSMISSION ON THE SELECTED
 (1)                                    LINE DID NOT DISTURB ANY UNSELECTED LOCATIONS IN THE MEMORIES. IF ALSO OPERATES
 (1)                                    IN "FLAG" MODE RATHER THAN USING INTERRUPTS. THE TEST SEQUENCE IS AS
 (1)                                    FOLLOWS:
 (1)
 (1)                                            1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
 (1)                                            2.  CLEAR BOTH THE "CAR" AND "BCR" MEMORIES
 (1)                                            3.  LOAD THE "BCR" MEMORY WITH ALL ONES (BYTE COUNT = -1)
 (1)                                            4.  ACTIVATE THE XMITTER ON THE SELECTED LINE
 (1)                                            5.  ACTIVATE TIMER TO WAIT FOR "XMIT DONE"
 (1)                                            6.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR AND
 (1)                                                REPEAT 1 THRU 5 UNTIL ALL SELECTED LINES TESTED
 (1)                                            7.  IF "XMIT DONE" SETS CHECK ALL LOCATIONS IN THE "BCR" MEMORY
 (1)                                                REPORT ANY UNSELECTED LINES NOT CONTAINING -1 AND THE SELECTED
 (1)                                                LINE IF IT DOES NOT CONTAIN 0
 (1)                                            8.  CHECK ALL LOCATIONS IN THE "CAR" MEMORY AND REPORT ANY UNSELECTED
 (1)                                                LOCATIONS NOT CONTAINING 0 AND THE SELECTED LINE IF IT DOES NOT
 (1)                                                CONTAIN +1.
 (1)                                            9.  REPEAT STEPS 1 THRU 8 UNTIL ALL SELECTED LINES TESTED.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 50] CALLED IF XMIT DONE TIMEOUT ERROR DETECTED.
 (1)                                    2.  [ERROR 51] CALLED IF "BCR" MEMORY ERROR DETECTED
 (1)                                    3.  [ERROR 51] CALLED IF "CAR" MEMORY ERROR DETECTED
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H     .   EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  ASSUMING TEST 32 RAN ERROR FREE THE PROBLEM IS MOST LIKELY THE:
 (1)                                            M7278 MODULE IF "BCR" ERRORS
 (1)                                            M7277 MODULE IF "CAR" ERRORS
 (1)
 (1)                                    KEY LOGIC:      (SAME AS TEST 32)
 (1)                                    **********
 (1)
 (1)                                    %
3727  010516  012767  010532  170364            MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
3728  010524  004767  012274            1$:     JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
3729  010530  000544                            BR      TST34           ;;BR IF DONE ALL SELECTED LINES
```

```
3730  010532  052711  004000          2$:   BIS     #BIT11,(R1)     ;CLEAR THE DH11
3731  010536  004767  012434                JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
3732  010542  004767  012472                JSR     PC,LOBCR        ;GO LOAD "BCR" MEMORY WITH ALL ONES
3733  010546  156711  014372                BISB    LINE,(R1)       ;SELECT THE LINE
3734  010552  012761  033500  000004        MOV     #33500,LPR(R1)  ;SET UP PARAMETERS
3735  010560  016761  013626  000012        MOV     LINMSK,BAR(R1)  ;ACTIVATE XMIT ON SELECTED LINE
3736
3737  010566  012767  000001  014376        MOV     #1,TIMEA        ;INIT TIMER A
3738  010574  005067  014374                CLR     TIMEB           ;INIT TIMER B
3739  010600  005711                  3$:   TST     (R1)            ;XMITTR DONE YET
3740  010602  100423                        BMI     4$              ;BR IF YES
3741  010604  004767  013342                JSR     PC,TIMEIT       ;CALL THE TIMER
3742  010610  000773                        BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                     ;AROUND THIS BRANCH IF TIME OUT OCCURS
3743
3744  010612  004767  013516                JSR     PC,SAPS         ;SAVE THE ERROR PSW
3745  010616  011103                        MOV     (R1),R3         ;GET THE WAS DATA
3746  010620  012704  100000                MOV     #BIT15,R4       ;SET UP S/B DATA
3747  010624  156704  014314                BISB    LINE,R4
3748  010630  010102                        MOV     R1,R2           ;MAKE REGADR = DEVADR
3749  010632  004767  012040                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3750  010636  004567  012254                JSR     R5,SUNUM        ;SET LINE NO. IN MSG
3751  010642  025144                        LINE
3752  010644  031076                        EMS0+53
3753  010646  104050                        ERROR   50              ;TIMED OUT AWAITING XMIT DONE ON SEL LINE
3754  010650  000725                        BR      1$              ;GO TRY THE NEXT LINE
3755
3756  010652  005067  170340          4$:   CLR     STMP7           ;INIT A LINE COUNTER
3757  010656  116711  170334          5$:   MOVB    STMP7,(R1)      ;SELECT LINE NO. IN "SCR"
3758  010662  012704  177777                MOV     #-1,R4          ;SET UP S/B DATA
3759  010666  016103  000010                MOV     BCR(R1),R3      ;GET THE WAS BYTE COUNT
3760  010672  126767  014246  170316        CMPB    LINE,STMP7      ;WAS THIS THE ACTIVE LINE ??
3761  010700  001001                        BNE     6$              ;BR IF NOT
3762  010702  005004                        CLR     R4              ;CHANGE S/B DATA TO 000000
3763  010704  020304                  6$:   CMP     R3,R4           ;WAS BYTE COUNT CORRECT ??
3764  010706  001416                        BEQ     7$              ;BR IF YES
3765
3766  010710  005067  170264                CLR     STMP0           ;SAVE THE ACTIVE LINE NO.
3767  010714  116767  014224  170256        MOVB    LINE,STMP0
3768  010722  016767  170270  170252        MOV     STMP7,STMP1     ;SAVE THE LINE NO. BEING CHECKED
3769  010730  010102                        MOV     R1,R2           ;SET UP REGADR = BCR REG ADDR
3770  010732  062702  000010                ADD     #BCR,R2
3771  010736  004767  012016                JSR     PC,SUER4        ;GO SET UP ERROR INFO
3772  010742  104051                        ERROR   51              ;BYTE COUNT INCORRECT
3773
3774  010744  005004                  7$:   CLR     R4              ;SET UP S/B DATA
3775  010746  016103  000006                MOV     CAR(R1),R3      ;GET THE WAS DATA
3776  010752  126767  014166  170236        CMPB    LINE,STMP7      ;IS THIS THE ACTIVE LINE
3777  010760  001001                        BNE     8$              ;BR IF NOT
3778  010762  005204                        INC     R4              ;BUMP THE CAR ADDRESS FOR ACTIVE LINE
3779  010764  020304                  8$:   CMP     R3,R4           ;CAR CONTENTS CORRECT ??
3780  010766  001416                        BEQ     9$              ;BR IF YES
3781
3782  010770  005067  170204                CLR     STMP0           ;SET UP ACT LINE NO.
```

```
3784  010774  116767  014144  170176          MOVB    LINE,$TMP0
3785  011002  016767  170210  170172          MOV     $TMP7,$TMP1     ;SAVE THE LINE NO. BEING CHECKED
3786  011010  010102                          MOV     R1,R2           ;SET UP REGADR
3787  011012  062702  000006                  ADD     #CAR,R2
3788  011016  004767  011736                  JSR     PC,SUER4        ;SET UP THE ERROR INFO
3789  011022  104051                          ERROR   51              ;CAR REG INCORRECT
3790
3791  011024  005267  170166          9$:     INC     $TMP7           ;GENERATE NEW LINE NO.
3792  011030  022767  000020  170160          CMP     #20,$TMP7       ;TESTED ALL LINES
3793  011036  001307                          BNE     5$              ;BR IF NOT
3794  011040  000631                          BR      1$              ;GO SELECT NEXT ACTIVE LINE
```

```
3797                                    ;*****************************************************************
 (3)                                    ;*TEST 34       TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
 (3)                                    ;*****************************************************************
 (2)  011042  000004                    TST34:  SCOPE
 (1)  011044  012767  000034  170162            MOV     #STN-1,STESTN   ;;SET TEST NUMBER IN  MAIL BOX
3798                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT WHEN "CHAR AVAIL" (BIT07 IN "SCR") SETS
 (1)                                    AS A RESULT OF XMITTING AND RECEIVING ONE CHARACTER (USING SILO MAINT MODE)
 (1)                                    IT CAUSES A RCVR INTR VIA THE PROPER VECTOR. THE TEST SEQUENCE IS AS
 (1)                                    FOLLOWS:
 (1)
 (1)                                            1.  SET UP THE RCVR VECTOR
 (1)                                            2.  LOCK OUT INTRS, RESET SP, AND CLEAR DH11
 (1)                                            3.  USE MAINT MODE TO LOAD A CHAR INTO THE SILO (DATA=125252)
 (1)                                            4.  CLEAR PSW TO ALLOW INTERRUPTS
 (1)                                            5.  ACTIVATE TIMER TO WAIT FOR INTR TO OCCUR
 (1)                                            6.  IF NO RCVR INTR OCCURS REPORT ERROR AND GO TO STEP 9
 (1)                                            7.  WHEN RCVR INTRS - CHECK SILO DATA FOR 125252 - IF NOT
 (1)                                                CORRECT REPORT ERROR AND GO TO STEP 9
 (1)                                            8.  CHECK THAT SILO FILL LEVEL=1 - IF NOT REPORT ERROR
 (1)                                            9.  RESET SP, CLEAR PSW, RESET VECTOR, AND GO TO TEST 35
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 13] IS CALLED TO REPORT RCVR TIMEOUT ERROR
 (1)                                    2.  [ERROR 52] IS CALLED TO REPORT SILO DATA INCORRECT
 (1)                                    3.  [ERROR  6] IS CALLED TO REPORT INCORRECT SILO FILL COUNT
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  IF NOT RCVR INTR OCCURS SUSPECT THE M7277, M7281, OR M7279 MODULES
 (1)                                    2.  IF SILO DATA OR FILL-ERRORS SUSPECT THE M7279 MODULE
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7279   SH1     SILO DATA MUX'S (74157'S)
 (1)                                                            SSR15 H         CR1
 (1)
 (1)                                                    SH2     DATA READY L            DV1
 (1)                                                            MRC 15 H                DL1
 (1)                                                            SILO MEMORY             (E13,E17,E8,E3) [3341'S]
 (1)                                                            LOAD SILO L             DJ2
 (1)                                                            5.068 MHZ (CLOCK)       DN1
 (1)                                                            SSR <13:00> H           (E20,E24)
 (1)
```

```
(1)                                                 M7289   SH6     RCV INT REQ H           DP1
(1)
(1)                                                 M7278   SH8     SSR <03:00> H
(1)                                                         SH7     SSR <07:00> H
(1)
(1)
(1)                                     %
(1)
3799  011052  012767  011074  170030                MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3800  011060  016703  013316                        MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
3801  011064  012723  011160                        MOV     #3$,(R3)+       ;GO TO 3$ ON RCVR INTERRUPT
3802  011070  116713  014044                        MOVB    DHRLVL,(R3)
3803  011074  004767  013220              1$:       JSR     PC,CHPS2        ;GO LOCK OUT INTRS
3804  011100  012706  001100                        MOV     #STACK,SP       ;RESET SP FOR ERROR LOOPS
3805  011104  012711  004000                        MOV     #BIT11,(R1)     ;CLEAR THE DH11
3806  011110  052761  100000  000016                BIS     #BIT15,SSR(R1)  ;SET SILO MAINT. BIT TO LOAD SILO
3807  011116  012711  000100                        MOV     #BIT06,(R1)     ;ENABLE CHAR. AVAIL INTERRUPT
3808  011122  004767  013156                        JSR     PC,CHPS1        ;GO CLEAR PSW
3809  011126  012703  001000                        MOV     #1000,R3        ;INIT TIMER
3810  011132  005303              2$:       DEC     R3              ;DEC TIMER
3811  011134  001376                        BNE     2$              ;BR IF NO TIMEOUT
3812
3813  011136  004767  013172                        JSR     PC,SAPS         ;SAVE THE ERROR PSW
3814  011142  011103                        MOV     (R1),R3         ;GET THE WAS DATA
3815  011144  012704  000300                        MOV     #300,R4         ;SET UP S/B DATA
3816  011150  004767  011522                        JSR     PC,SUER2A       ;GO SET UP ERROR INFO
3817  011154  104013                        ERROR   13              ;CHAR AVAIL FAILED TO SET ON TIME
3818  011156  000436                        BR      5$              ;ESCAPE FROM THIS TEST - CATASTROPHIC ERROR
3819
3820  011160  016105  000016              3$:       MOV     SSR(R1),R5      ;SAVE THE SILO STATUS REG.
3821  011164  016103  000002                        MOV     NRC(R1),R3      ;GET THE WAS DATA
3822  011170  012704  125252                        MOV     #125252,R4      ;SET UP S/B DATA
3823  011174  020304                        CMP     R3,R4           ;WAS = S/B = 125252 ??
3824  011176  001410                        BEQ     4$              ;BR IF IT IS
3825
3826  011200  004767  013130                        JSR     PC,SAPS         ;SAVE THE ERROR PSW
3827  011204  062702  000002                        ADD     #NRC,R2         ;SET  UP REGADR
3828  011210  004767  011462                        JSR     PC,SUER2A       ;GO SET UP  ERROR INFO
3829  011214  104052                        ERROR   52              ;DATA COMPARE ERROR
3830  011216  000416                        BR      5$              ;GET OUT
3831
3832  011220  010503              4$:       MOV     R5,R3           ;NOW GET THW SILO STATUS REG AGAIN
3833  011222  042703  140377                        BIC     #140377,R3      ;CLR OUT JUNK
3834  011226  012704  000400                        MOV     #400,R4         ;SET UP S/B DATA
3835  011232  020304                        CMP     R3,R4           ;SSR CHAR COUNT = 1 ??
3836  011234  001407                        BEQ     5$              ;BR IF IT IS
3837
3838  011236  004767  013072                        JSR     PC,SAPS         ;SAVE THE ERROR PSW
3839  011242  062702  000016                        ADD     #SSR,R2         ;SET UP REGADR
3840  011246  004767  011424                        JSR     PC,SUER2A       ;SET UP ERROR INFO
3841  011252  104006                        ERROR   6               ;SSR COUNT NOT CORRECT
3842
3843  011254  012706  001100              5$:       MOV     #STACK,SP       ;RESET THE STACK POINTER
3844  011260  004767  013020                        JSR     PC,CHPS1        ;GO CLEAR PSW
3845  011264  005011                        CLR     (R1)            ;RESET I.E. BIT
```

```
3846  011266  016703  013110          MOV     DHVCT,R3        ;GET FIRST VECTOR ADDR
3847  011272  010313                  MOV     R3,(R3)         ;RESTORE TRAP CATCHER
3848  011274  062723  000002          ADD     #2,(R3)+
3849  011300  005013                  CLR     (R3)
3850
```

```
3853                                    ;*****************************************************************
 (3)                                    ;*TEST 35       TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
 (3)                                    ;*****************************************************************
 (2)  011302  000004                    TST35:  SCOPE
 (1)  011304  012767  000035  167722            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3854                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    ***************
 (1)
 (1)                                            THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS UP CORRECTLY
 (1)                                    WHEN ALL COUNTS (0-77) ARE TESTED BY LOADING THE SILO USING
 (1)                                    MAINT MODE. THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1.  INIT "$TMP7" TO START WITH A COUNT=01
 (1)                                            2.  CLEAR THE DH11
 (1)                                            3.  LOAD THE SILO WITH 125252'S IN MAINT MODE
 (1)                                                UNTIL # OF WORDS INDICATED BY THE COUNT ARE LOADED
 (1)                                            4.  AFTER LOADING REQUIRED COUNT CHECK THAT FILL LEVEL
 (1)                                                BITS (SSR<13:08>) EQUAL COUNT - IF NOT REPORT ERROR
 (1)                                            5.  INCREMENT COUNT IN "$TMP7" AND REPEAT 1 THRU 4
 (1)                                                UNTIL ALL COUNTS (01-77) HAVE BEEN TESED
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  [ERROR 6] IS CALLED TO REPORT SILO FILL LEVEL ERRORS
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  FAILURES IN THIS TEST MOST LIKELY INDICATE A BAD M7279 MODULE
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7279   SH2     SSR <13:08>
 (1)                                                            LOAD SILO L DJ2
 (1)                                                            5.068 MHZ (CLOCK) DN1
 (1)                                                            DATA READY L DV1
 (1)
 (1)                                    %
3855  011312  010102                            MOV     R1,R2           ;MAKE REGADR = SSR
3856  011314  062702  000016                    ADD     #SSR,R2
3857  011320  012767  000001  167670            MOV     #1,$TMP7        ;START WITH COUNT OF 1
3858  011326  012711  004000           1$:      MOV     #BIT11,(R1)     ;CLEAR THE DH11
3859  011332  016705  167660                    MOV     $TMP7,R5        ;SAVE CHARACTER COUNT BEING TESTED
3860  011336  005004                            CLR     R4              ;INIT A CHAR COUNTER
3861  011340  052712  100000           2$:      BIS     #BIT15,(R2)     ;SET THE SILO MAINT BIT
3862  011344  012703  001000                    MOV     #1000,R3        ;INIT A TIMER
3863  011350  005303                    3$:     DEC     R3              ;STALL TO ALLOW TIME TO LOAD SILO
3864  011352  001376                            BNE     3$
```

```
3865  011354  042712  100000                  BIC     #BIT15,(R2)     ;CLEAR SILO MAINT. BIT
3866  011360  005204                          INC     R4              ;COUNT A CHAR LOADED
3867  011362  005305                          DEC     R5              ;DECREMENT TEST COUNT
3868  011364  001365                          BNE     2$              ;BR UNTIL WE'VE LOADED THE TEST COUNT
3869
3870  011366  011203                          MOV     (R2),R3         ;SET THE WAS COUNT
3871  011370  042703  140377                  BIC     #140377,R3      ;CLR JUNK BITS
3872  011374  000304                          SWAB    R4              ;SET UP S/B DATA
3873  011376  020304                          CMP     R3,R4           ;TEST COUNT = SILO COUNTER ?
3874  011400  001406                          BEQ     4$              ;BR IF YES
3875
3876  011402  004767  011264          JSR     PC,SUER2                ;GO SET UP ERROR INFO
3877  011406  012767  011326  167474          MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
3878  011414  104006                          ERROR   6               ;SSR FAILED TO UP-COUNT CORRECTLY
3879
3880  011416  005267  167574          4$:     INC     STMP7           ;INCREMENT TO NEXT COUNT TO TEST
3881  011422  022767  000100  167566          CMP     #100,STMP7      ;MAXIMUM COUNT ??
3882  011430  001336                          BNE     1$              ;BR IF NOT
3883
```

```
                                        ;*****************************************************************
                                        ;*TEST 36       TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
                                        ;*****************************************************************
(2)  011432  000004                     TST36:  SCOPE
(1)  011434  012767  000036  167572             MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                        .REM    X
                                        TEST ABSTRACT:
                                        **************

                                                THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS DOWN PROPERLY
                                        WHEN WORDS ARE READ FROM THE SILO. ALL COUNTS FROM 77-00 ARE
                                        TESTED. THE TEST SEQUENCE IS AS FOLLOWS:

                                                1.  INIT "$TMP7" TO START WITH A COUNT OF 1
                                                2.  CLEAR THE DH11 AND FILL SILO WITH 64. WORDS
                                                3.  READ THE NO. OF WORDS SPECIFIED BY COUNT
                                                4.  CHECK THAT FILL LEVEL=64. MINUS (COUNT) - REPORT ERRORS
                                                5.  INCREMENT "$TMP7" AND REPEAT 2 THRU 4 UNTIL ALL
                                                    COUNTS TESTED.

                                        ERRORS:
                                        *******

                                        1.  (ERROR 6) IS CALLED TO REPORT SILO FILL LEVEL ERRORS

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:          (REFER TO TEST 35)
                                        ******

                                        KEY LOGIC:      (REFER TO TEST 35)
                                        **********

                                        %
3888  011442  010102                            MOV     R1,R2           ;SET UP REGADR
3889  011444  062702  000016                    ADD     #SSR,R2
3890  011450  012767  000001  167540            MOV     #1,$TMP7        ;START WITH COUNT = 1
3891  011456  012711  004000           1$:      MOV     #BIT11,(R1)     ;CLR THE DH11
3892  011462  012705  000100                    MOV     #100,R5         ;TEST COUNT SHOULD BE 64(10) MINUS
3893  011466  166705  167524                    SUB     $TMP7,R5        ;THE NO. OF CHARS READ
3894  011472  012703  000100                    MOV     #100,R3         ;COUNTER USED TO FILL SILO
3895  011476  012704  001000           2$:      MOV     #1000,R4        ;INIT TIMER
3896  011502  052712  100000                    BIS     #BIT15,(R2)     ;SET SILO MAINT. BIT
3897  011506  005304                    3$:      DEC     R4              ;STALL TO ALLOW SILO TO LOAD
3898  011510  001376                            BNE     3$
3899  011512  042712  100000                    BIC     #BIT15,(R2)     ;CLEAR THE SILO MAINT BIT
3900  011516  005303                            DEC     R3              ;COUNT ONE CHAR LOADED
3901  011520  001366                            BNE     2$              ;BR UNTIL ALL LOADED
3902
3903  011522  016703  167470                    MOV     $TMP7,R3        ;INIT COUNTER FOR READING SILO
3904  011526  012704  001000           4$:      MOV     #1000,R4        ;INIT TIMER
3905  011532  005761  000002                    TST     NRC(R1)         ;READ THE SILO
3906  011536  005304                    5$:      DEC     R4              ;GIVE IT TIME TO SETTLE
```

```
3907  011540  001376                        BNE     5$
3908  011542  005303                        DEC     R3              ;COUNT ONE READ
3909  011544  001370                        BNE     4$              ;BR UNTIL WE'VE READ TEST COUNT
3910
3911  011546  011203                        MOV     (R2),R3         ;GET THE WAS DATA
3912  011550  042703  140377                BIC     #140377,R3      ;CLR JUNK BITS
3913  011554  010504                        MOV     R5,R4           ;SET UP S/B DATA
3914  011556  000304                        SWAB    R4
3915  011560  020304                        CMP     R3,R4           ;DID IT DOWN COUNT OK ??
3916  011562  001406                        BEQ     6$              ;BR IF YES
3917
3918  011564  004767  011102                JSR     PC,SUBR2        ;GO SET UP ERROR INFO
3919  011570  012767  011456  167312        MOV     #1$,$LPERR      ;SET UP ERROR LOOP RETURN
3920  011576  104006                        ERROR   6               ;SILO STATUS REG. DOWN-COUNTED INCORRECTLY
3921
3922  011600  005267  167412        6$:     INC     $TMP7           ;UPDATE COUNT
3923  011604  022767  000101  167404        CMP     #101,$TMP7      ;TESTED ALL COUNTS ??
3924  011612  001321                        BNE     1$              ;BR IF NOT
3925
3926
```

```
3929                                    ;*****************************************************************
 (3)                                    ;*TEST 37       TEST SILO ALARM LEVEL FOR  COUNTS 0,1,2,4,8,16, AND 32
 (3)                                    ;*****************************************************************
 (2)  011614  000004                    TST37:  SCOPE
 (1)  011616  012767  000037  167410            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
3930                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT THE SILO ALARM LEVEL WORKS PROPERLY FOR
 (1)                                    INTEGRAL POWER OF 2 COUNTS (0, 1, 2, 4, 8, 16, AND 32). THE TEST SEQUENCE
 (1)                                    IS AS FOLLOWS:
 (1)
 (1)                                            1.  INIT "$TMP7" TO START WITH ALARM LEVEL OF 000
 (1)                                            2.  CLEAR THE DH11 AND LOAD THEN SILO WITH THAT NO. OF WORDS
 (1)                                                THAT IS ONE GREATER THAN THE ALARM LEVEL.
 (1)                                            3.  VERIFY THAT "DATA READY" DOES NOT SET UNTIL THE FILL
 (1)                                                LEVEL EXCEEDS THE ALARM LEVEL.
 (1)                                            4.  REPORT ERRORS IF:
 (1)                                                A.  "READY" SETS TOO SOON
 (1)                                                B.  "READY" SETS TOO LATE
 (1)                                            5.  SHIFT "$TMP7" LEFT TO GENERATE NEXT POWER OF 2 LEVEL
 (1)                                            6.  REPEAT 2 THRU 5 UNTIL ALL 7 TEST LEVELS CHECKED
 (1)
 (1)                                            NOTE:   FOR (A) ABOVE IF "READY" SETS JUST ONE WORK TOO
 (1)                                                    SOON IT IS ALLOWED BY ANYTHING GREATER RESULTS IN
 (1)                                                    AN ERROR MESSAGE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1.  (ERROR 6) IS CALLED TO REPORT BOTH TYPES OF ERRORS OUTLINED
 (1)                                        IN 4(A,B) ABOVE
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1.  ERRORS IN THIS TEST ONLY INDICATE BAD COMPARATOR CHIP (E23 OR E19)
 (1)                                        ON THE M7279 - SH2
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7279   SH2     E19 - PIN 5 (COMPARATOR)
 (1)                                            ALSO SAME LOGIC AS TEST 35
 (1)
 (1)                                    %
3931  011624  012767  011644  167256            MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
3932  011632  010102                            MOV     R1,R2           ;SET UP REGADR
3933  011634  062702  000016                    ADD     #$SR,R2
3934  011640  005067  167352                    CLR     $TMP7           ;START WITH LEVEL 00
```

```
3935  011644  012711  004000          1$:   MOV   #BIT11,(R1)    ;CLEAR THE DH11
3936  011650  016705  167342                MOV   $TMP7,R5       ;SAVE IT IN R5
3937  011654  010512                       MOV   R5,(R2)        ;SET ALARM LEVEL IN SSR
3938  011656  005205                       INC   R5             ;LOAD ONE MORE THAN FILL LEVEL
3939  011660  052712  100000          2$:   BIS   #BIT15,(R2)    ;SET SILO MAINT. TO LOAD A CHAR
3940  011664  012703  001000                MOV   #1000,R3       ;INIT STALL TIMER
3941  011670  005303                3$:   DEC   R3             ;WAIT FOR SILO TO SETTLE
3942  011672  001376                       BNE   3$             ;BR TIL R3 GOES TO 000000
3943  011674  042712  100000                BIC   #BIT15,(R2)    ;CLR THE SILO MAINT BIT
3944  011700  005305                       DEC   R5             ;COUNT ONE LOADED
3945  011702  105711                       TSTB  (R1)           ;CHAR AVAIL SET YET
3946  011704  100412                       BMI   4$             ;BR IF IT IS
3947  011706  005705                       TST   R5             ;SHOULD IT BE ??
3948  011710  001363                       BNE   2$             ;BR IF NOT
3949
3950  011712  004767  012416                JSR   PC,SAPS        ;SAVE THE ERROR PSW
3951  011716  004767  000042                JSR   PC,5$          ;GO SET UP S/B DATA
3952  011722  004767  010750                JSR   PC,SUER2A      ;GO SET UP ERROR INFO
3953  011726  104006                       ERROR 6              ;SILO ALARM LEVEL FAILED AT SELECTED COUNT
3954  011730  000426                       BR    6$             ;GO CHECK NEXT COUNT
3955
3956  011732  005705                4$:   TST   R5             ;SHOULD IT HAVE BEEN SET (CHAR AVAIL)
3957  011734  001424                       BEQ   6$             ;BR IF YES
3958
3959  011736  004767  012372                JSR   PC,SAPS        ;SAVE THE ERROR PSW
3960  011742  022705  000001                CMP   #1,R5          ;IS IT OFF BY ONLY ONE ??
3961  011746  001417                       BEQ   6$             ;BR IF YES - WE'LL ALLOW HIM THIS
3962  011750  004767  000010                JSR   PC,5$          ;GO SET UP S/B DATA
3963  011754  004767  010716                JSR   PC,SUER2A      ;GO SET UP ERROR INFO
3964  011760  104006                       ERROR 6              ;SILO ALARM LEVEL FAILED
3965  011762  000411                       BR    6$             ;GO CHECK NEXT COUNT
3966
3967  011764  011203                5$:   MOV   (R2),R3        ;GET WAS DATA
3968  011766  016704  167224                MOV   $TMP7,R4       ;SET UP THE S/B DATA
3969  011772  005204                       INC   R4
3970  011774  000304                       SWAB  R4
3971  011776  105004                       CLRB  R4
3972  012000  156704  167212                BISB  $TMP7,R4
3973  012004  000207                       RTS   PC             ;RETURN TO SET UP AND REPORT ERROR
3974
3975  012006  005767  167204          6$:   TST   $TMP7          ;COUNT AT ZERO
3976  012012  001002                       BNE   7$             ;BR IF NOT
3977  012014  000261                       SEC                  ;SET THE "C" BIT
3978  012016  000401                       BR    8$             ;GO SET UP COUNT
3979  012020  000241                7$:   CLC                  ;CLEAR THE "C" BIT
3980  012022  006167  167170          8$:   ROL   $TMP7          ;SHIFT POWER OF TWO BIT
3981  012026  032767  000100  167162       BIT   #BIT6,$TMP7    ;DONE ALL POWERS ??
3982  012034  001703                       BEQ   1$             ;BR IF NOT
```

```
3985                                    ;*****************************************************************
 (3)                                    ;*TEST 40       VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL LINES
 (3)                                    ;*****************************************************************
 (2)  012036  000004                    TST40:  SCOPE
 (1)  012040  012767  000040  167166            MOV     #STN-1,STESTN   ;;SET TEST NUMBER IN  MAIL BOX
3986                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT THE STORAGE OVERFLOW BIT (SCR14) SETS
 (1)                                    AND CLEARS PROPERLY FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS
 (1)                                    FOLLOWS:
 (1)
 (1)                                            1. SET UP THE ERROR RETURN
 (1)                                            2. SELECT A LINE NUMBER TO TEST - GO TO NEXT TEST IF ALL
 (1)                                               SELECTED LINES HAVE BEEN TESTED.
 (1)                                            3. PRIME THE SELECTED LINE TO XMIT 65(10) CHARS.
 (1)                                            4. ACTIVATE THE SELECTED LINE AND WAIT FOR STORAGE OVERFLOW
 (1)                                               TO SET (SCR14=1)
 (1)                                            5. IF SCR14 FAILS TO SET ON TIME - REPORT ERROR AND THEN
 (1)                                               CONTINUE WITH THE NEXT LINE (STEP 2).
 (1)                                            6. IF IT SETS OK - READ THE "NRC" REG TWICE TO EMPTY TWO
 (1)                                               WORDS FROM THE SILO.
 (1)                                            7. AFTER A BRIEF STALL, VERIFY THAT SCR14 HAS CLEARED -
 (1)                                               IF NOT REPORT ERROR AND CONTINUE WITH NEXT LINE (STEP2)
 (1)                                            8. IF IT CLEARS OK, VERIFY THAT THE FILL COUNT (SSR<15:08>)
 (1)                                               CONTAINS A 77(8) - IF NOT REPORT ERROR AND CONTINUE
 (1)                                               WITH NEXT LINE  (STEP 2).
 (1)                                            9. REPEAT STEPS 2 THRU 8 UNTIL ALL SELECTED LINES HAVE BEEN
 (1)                                               TESTED.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1. [ERROR  57] IS CALLED TO REPORT ALL ERRORS DETECTED.
 (1)
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A  H       EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1. PROBLEM IS MOST LIKELY ON THE M7289 MODULE (SHH) OR SOME
 (1)                                       SIGNAL FEEDING THIS LOGIC.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7289   SHH     STORAGE OVERFLOW L       E43-12
 (1)                                                            READY IN PULSE H         E40-11
 (1)                                                            UC1 MASTER DA H          BH2
 (1)                                                            UC2 MASTER DA H          BD2
 (1)
```

```
(1)                                         %
3987  012046  012767  012062  167034        MOV     #2$,$LPERR      ;SET UP ERROR RETURN
3988  012054  004767  010744          1$:   JSR     PC,$ELINE       ;GO SELECT A LINE NO. TO TEST
3989  012060  000535                        BR      TST41           ;;BR IF DONE ALL SELECTED LINES
3990  012062  012711  004000          2$:   MOV     #3IT11,(R1)     ;CLEAR THE DH11
3991  012066  010102                        MOV     R1,R2           ;MAKE REGADR = DEVADR
3992  012070  116711  013050                MOVB    LINE,(R1)       ;SELECT A LINE
3993  012074  005061  000006                CLR     CAR(R1)         ;SET UP CURRENT ADDRESS
3994  012100  012761  177677  000010        MOV     #-65.,BCR(R1)   ;SET UP BYTE COUNT
3995  012106  012761  033500  000004        MOV     #33[illegible]0,LPR(R1) ;SET UP LPR: 9600 BAUD,5 BIT CHARS
3996  012114  016761  012272  000012        MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
3997
3998  012122  012767  000001  013042        MOV     #1,TIMEA        ;INIT TIMERS
3999  012130  005067  013040                CLR     TIMEB
4000  012134  032711  040000          3$:   BIT     #BIT.4,(R1)     ;STOREGE OVERFLOW YET ??
4001  012140  001024                        BNE     4$              ;BR IF YES YOU SHOULD GET IT
4002  012142  004767  012004                JSR     PC,TIMEIT       ;CALL TIMER
4003  012146  000772                        BR      3$              ;BR IF NO TIME OUT
4004
4005  012150  004767  012160                JSR     PC,SAPS         ;GO SAVE PSW
4006  012154  012704  040000                MOV     #BIT14,R4       ;SET UP S/B DATA
4007  012160  156704  012760                BISB    LINE,R4
4008  012164  011103                        MOV     (R1),R3         ;SET UP WAS DATA
4009  012166  042703  137760                BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
4010  012172  004767  010500                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4011  012176  004567  010714                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4012  012202  025144                        LINE
4013  012204  031563                        EMS7+44
4014  012206  104057                        ERROR   57              ;STORAGE OVERFLOW FAILED TO SET
4015  012210  000721                        BR      1$              ;GO TRY NEXT LINE
4016
4017  012212  016167  000002  166776  4$:   MOV     NRC(R1),$TMP7   ;READ THE SILO
4018  012220  016167  000002  166770        MOV     NRC(R1),$TMP7   ;READ IT AGAIN
4019  012226  012705  001000                MOV     #1000,R5        ;INIT STALL COUNTER
4020  012232  005305                  41$:  DEC     R5              ;COUNT TIMER
4021  012234  001376                        BNE     41$             ;BR IF NO TIMEOUT
4022  012236  032711  040000                BIT     #BIT14,(R1)     ;DID OVERFLOW GO AWAY ?
4023  012242  001420                        BEQ     5$              ;BR IF YES
4024
4025  012244  004767  012064                JSR     PC,SAPS         ;GO SAVE THE PSW
4026  012250  005004                        CLR     R4              ;SET UP S/B DATA
4027  012252  156704  012666                BISB    LINE,R4
4028  012256  011103                        MOV     (R1),R3         ;SET UP WAS DATA
4029  012260  042703  137760                BIC     #137760,R3      ;CLEAR UNINTERESTING BITS
4030  012264  004767  010406                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4031  012270  004567  010622                JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
4032  012274  025144                        LINE
4033  012276  031563                        EMS7+44
4034  012300  104057                        ERROR   57              ;STORAGE BIT FAILED TO CLEAR
4035  012302  000664                        BR      1$              ;GO TRY NEXT LINE
4036
4037  012304  122761  000077  000017  5$:   CMPB    #77,SSR+1(R1)   ;WAS IT REALLY 65. ??
4038  012312  001660                        BEQ     1$
4039
```

```
 4040  012314  004767  012014          JSR     PC,SAPS         ;GO SAVE PSW
 4041  012320  012704  037400          MOV     #37400,R4       ;SET UP S/B DATA
 4042  012324  062702  000016          ADD     #SSR,R2         ;SET UP REGADR
 4043  012330  016103  000016          MOV     SSR(R1),R3      ;SAVE WAS DATA
 4044  012334  004767  010336          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4045  012340  004567  010552          JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
 4046  012344  025144                  LINE
 4047  012346  031563                  EMS7+44
 4048  012350  104057                  ERROR   57              ;READING SILO FAILED TO DEC SSR OR
                                                               ;STORAGE OVFL SET AT WRONG COUNT
 4050  012352  000640                  BR      1$              ;GO TRY NEXT LINE
```

```
4053                                ;*****************************************************************
 (3)                                ;*TEST 41       TRANSMITTER TIMING TEST - ALL LINES - ALL SPEEDS
 (3)                                ;*****************************************************************
 (2)  012354  000004                TST41:  SCOPE
 (1)  012356  012767  000041  166650        MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
4054                                .REM    %
 (1)                                TEST ABSTRACT:
 (1)                                **************
 (1)
 (1)                                        THIS TEST PERFORMA A "RELATIVE" TIMING TEST FOR ALL BAUD RATES
 (1)                                ON ALL SELECTED LINES. IT DOES NOT MEASURE ABSOLUTE TIMES BUT
 (1)                                SIMPLY VERIFIES THAT EACH SUCCESSIVE SPEED FROM 50 TO 9600
 (1)                                BAUD IS FASTER THAN THE PREVIOUS SPEED. THE TEST SEQUENCE IS AS
 (1)                                FOLLOWS:
 (1)
 (1)                                        1.  SELECT A LINE # TO TEST (AS DEFINED BY "LINSEL:")
 (1)                                        2.  INIT "$TMP7" TO START WITH 50 BAUD AND A RELATIVE
 (1)                                            TIMER "TIMEC" TO -1 (177777)
 (1)                                        3.  CLEAR THE DH11 AND ACTIVATE SELECTED LINE TO TRANSMIT
 (1)                                            THREE CHARS.
 (1)                                        4.  ACTIVATE TIMER TO UPDATE "TIMEB" THE LINE SPEED TIMER
 (1)                                        5.  IF "XMIT DONE" FAILS TO SET ON TIME - REPORT ERROR
 (1)                                            AND REPEAT 3 THRU 4 UNTIL ALL SPEEDS CHECKED - THEN
 (1)                                            REPEAT 1 THRU 5 UNTIL ALL LINES CHECKED
 (1)                                        6.  IF "XMIT DONE" SETS VERIFY [TIMEB] LESS THAN [TIMEC]
 (1)                                            IF NOT REPORT ERROR - MAKE [TIMEC]=[TIMEB] AND
 (1)                                            REPEAT 3 THRU 5 UNTIL ALL SPEEDS CHECKED
 (1)                                        7.  REPEAT 1 THRU 6 FOR ALL SELECTED LINES.
 (1)
 (1)                                ERRORS:
 (1)                                *******
  1)
 (1)                                1.  [ERROR 53] IS CALLED TO REPORT XMIT TIMEOUT ERRORS
 (1)                                2.  [ERROR 17] IS CALLED TO REPORT TIMING ERRORS
 (1)
 (1)                                SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                *****
 (1)
 (1)                                DEBUG:
 (1)                                ******
 (1)
 (1)                                1.  IF ALL LINES FAIL ON ALL SPEEDS SUSPECT THE CLOCK MODULE M4540
 (1)                                2.  IF ALL LINES FAIL ON JUST ONE SPEED (THE SAME ONE) SUPECT EITHER THE
 (1)                                    CLOCK MODULE OR THE M7288 MODULE (TIMING SELECT MUXES)
 (1)                                3.  IF JUST ONE LINE FAILS SUSPECT EITHER THE UART MODULE (M7280
 (1)                                    EITHER FOR LINES <15:08> OR <07:00> OR THE M7288 MODULE
 (1)
 (1)                                KEY LOGIC:
 (1)                                **********
 (1)
 (1)                                        M4540   SH2             <9600:50> BAUD SIGNALS
 (1)                                        M7288   SH3             BOT AND TOP BUF CLOCK SIGNALS
 (1)                                                SH4,6,8,OR 10   TX CLOCK NN L SIGNALS
 (1)                                        M7280   TBMT LINE "N" SIGNALS ON UART PIN 22
```

```
                                                        TX CLOCK LINE "N" SIGNALS ON UART PIN 40
                                        %
012364  012767  012414  166516          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
012372  004767  010426          1$:     JSR     PC,$ELINE       ;GO SELECT A LINE TO TEST
012376  000534                          BR      TST42           ;BR IF TESTED ALL SELECTED LINES
012400  012767  002100  166610          MOV     #2100,$TMP7     ;INIT TI START WITH LOWEST SPEED
012406  012767  177777  012562          MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
012414  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE CH11
012420  156711  012520          3$:     BISB    LINE,(R1)       ;SELECT IT IN THE SCR
012424  012761  177775  000010          MOV     #-3,BCR(R1)     ;SET BYTE COUNT TO XFER 3 CHARS
012432  005061  000006                  CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
012436  016761  166554  000004          MOV     $TMP7,LPR(R1)   ;SELECT A XMIT SPEED
012444  016761  011742  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER

012452  012767  000001  012512          MOV     #1,TIMEA        ;INIT TIMER A
012460  005067  012510                  CLR     TIMEB           ;INIT TIMER B
012464  005711                  4$:     TST     (R1)            ;XMITTR DONE SET YET ?
012466  100437                          BMI     5$              ;BR IF YES
012470  004767  011456                  JSR     PC,TIMEIT       ;CALL THE TIMER
012474  000773                          BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
                                                                ;AROUND THIS BRANCH IF TIME OUT OCCURS

012476  016767  166514  166474          MOV     $TMP7,$TMP0     ;SAVE AND SET UP THE SPEED CODE
012504  000367  166470                  SWAB    $TMP0
012510  006267  166464                  ASR     $TMP0
012514  006267  166460                  ASR     $TMP0
012520  042767  177760  166452          BIC     #177760,$TMP0
012526  011103                          MOV     (R1),R3         ;GET THE WAS DATA
012530  042703  000200                  BIC     #BIT07,R3       ;CLEAR UNINTERESTING BITS
012534  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
012536  012704  100000                  MOV     #BIT15,R4       ;SET UP S/B DATA
012542  156704  012376                  BISB    LINE,R4
012546  004767  010124                  JSR     PC,$UER2A       ;GO SET UP ERROR INFO
012552  004567  010340                  JSR     R5,$UNUM        ;GO SET LINE NO. IN MSG
012556  025144                          LINE
012560  031076                          EMSG+53
012562  104053                          ERROR   53              ;TIMED OUT WAITING FOR XMIT DONE
012564  000426                          BR      8$              ;GO TEST NEXT SPEED

012566  016703  012402          5$:     MOV     TIMEB,R3        ;GET THE WAS COUNT
012572  016704  012400                  MOV     TIMEC,R4        ;GET LASTR CHECK COUNT
012576  020304                          CMP     R3,R4           ;COMPARE RELATIVE TIMES
012600  103420                          BLO     8$              ;BR IF THIS SPEED FASTER THAN LAST
                                                                ;SPEED TESTED

012602  004767  011526          7$:     JSR     PC,$APS         ;SAVE THE ERROR PSW
012606  016702  166404                  MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
012612  000302                          SWAB    R2
012614  006202                          ASR     R2
012616  006202                          ASR     R2
012620  042702  177760                  BIC     #177760,R2      ;STRIP AWAY ALL JUNK
012624  004767  010046                  JSR     PC,$UER2A       ;GO SET UP ERROR INFO
012630  004567  010262                  JSR     R5,$UNUM        ;GO PUT LINE NO. IN MSG
```

```
 4106  012634  025141                          LINE
 4107  012636  025141                          EM17+41
 4108  012640  104017                          ERROR   17              ;TRANSMITTER SPEED INCORRECT
 4109
 4110  012642  016767  012326  012326  BS:     MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
 4111  012650  062767  002100  166340          ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
 4112  012656  022767  035600  166332          CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
 4113  012664  001253                          BNE     2$              ;BR IF NOT
 4114  012666  000641                          BR      1$              ;GO TEST NEXT LINE
 4115
```

```
4118                                    ;*****************************************************************
(3)                                     ;*TEST 42       RECEIVER TIMING TEST - ALL LINES - ALL SPEEDS
(3)                                     ;*****************************************************************
(2)  012670  000004                     TST42:  SCOPE
(1)  012672  012767  000042  166334             MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
4119                                    .REM    X
(1)                                     TEST ABSTRACT:
(1)                                     **************
(1)
(1)                                             THIS TEST ID IDENTICAL TO TES 40 EXCEPT IT WAITS FOR "DATA READY"
(1)                                     TO CHECK RECEIVER TIMING. THE SEQUENCE IS SIMILAR AND THE SAME TIMERS
(1)                                     ARE USED FOR ERROR CHECKING.
(1)
(1)                                     ERRORS:
(1)                                     *******
(1)
(1)                                     1.  [ERROR 54] IS CALLED TO REPORT RCVR TIMEOUT ERRORS
(1)                                     2.  [ERROR 20] IS CALLED TO REPORT RCVR TIMING ERRORS
(1)
(1)                                     SYNC:   M7277   SH3     INIT A H        EF2
(1)                                     *****
(1)
(1)                                     DEBUG:          (SAME AS TEST 40)
(1)                                     ******
(1)
(1)                                     KEY LOGIC:      (SAME AS TEST 40 PLUS)
(1)                                     **********
(1)
(1)                                             M7288   SH5,7,9,11       RX CLOCK NN L SIGNALS
(1)                                             M7280   BUF DA LINE "N" UART PIN 19
(1)                                                     RX CLOCK LINE "N" UART PIN 17
(1)
(1)                                     %
4120  012700  012767  012730  166202        1$:     MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4121  012706  004767  010112                        JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4122  012712  000532                                BR      TST43           ;;BR IF TESTED ALL SELECTED LINES
4123  012714  012767  002100  166274                MOV     #2100,STMP7     ;INIT TO START WITH LOWEST SPEED
4124  012722  012767  177777  012246                MOV     #-1,TIMEC       ;INIT RELATIVE TIME CHECKER
4125  012730  012711  004000                2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
4126  012734  156711  012204                3$:     BISB    LINE,(R1)       ;SELECT IT IN THE SCR
4127  012740  012761  177777  000010                MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO XFER 1 CHAR
4128  012746  005061  000006                        CLR     CAR(R1)         ;GET TEST DATA STARTING AT LOC. 0
4129  012752  016761  166240  000004                MOV     STMP7,LPR(R1)   ;SELECT A XMIT SPEED
4130  012760  016761  011426  000012                MOV     LINMSK,BAR(R1)  ;ACTIVATE THE TRANSMITTER
4131
4132  012766  012767  000001  012176                MOV     #1,TIMEA        ;INIT TIMER A
4133  012774  005067  012174                        CLR     TIMEB           ;INIT TIMER B
4134  013000  105711                        4$:     TSTB    (R1)            ;RCVR DONE YET ??
4135  013002  100435                                BMI     5$              ;BR IF YES
4136  013004  004767  011142                        JSR     PC,TIMEIT       ;CALL THE TIMER
4137  013010  000773                                BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
4138                                                                          ;AROUND THIS BRANCH IF TIME OUT OCCURS
4139
4140  013012  016767  166200  166160                MOV     STMP7,STMP0     ;SAVE AND SET UP THE SPEED CODE
```

```
4141  013020  006367  166154                    ASL     $TMP0
4142  013024  006367  166150                    ASL     $TMP0
4143  013030  000367  166144                    SWAB    $TMP0
4144  013034  042767  177760  166136           BIC     #177760,$TMP0
4145  013042  011103                            MOV     (R1),R3         ;GET THE WAS DATA
4146  013044  010102                            MOV     R1,R2           ;MAKE REGADR = DEVADR
4147  013046  012704  100200                    MOV     #BIT15+BIT07,R4 ;SET UP S/B DATA
4148  013052  156704  012066                    BISB    LINE,R4
4149  013056  004767  007614                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4150  013062  004567  010030                    JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
4151  013066  025144                            LINE
4152  013070  026720                            EM22+51
4153  013072  104054                            ERROR   54              ;TIMED OUT WAITING FOR CHAR AVAIL
4154  013074  000426                            BR      8$              ;GO TEST NEXT SPEED
4155
4156  013076  016703  012072          5$:       MOV     TIMEB,R3        ;GET THE WAS COUNT
4157  013102  016704  012070                    MOV     TIMEC,R4        ;GET THE CHECK COUNT
4158  013106  020304                            CMP     R3,R4           ;COMPARE RELATIVE TIMES
4159  013110  103420                            BLO     8$              ;BR IF TIME INDICATES THIS SPEED FASTER
4160                                            ;THAN LAST SPEED
4161
4162  013112  004767  011216          7$:       JSR     PC,SAPS         ;SAVE THE ERROR PSW
4163  013116  016702  166074                    MOV     $TMP7,R2        ;GET SPEED CODE AND RIGHT JUSTIFY
4164  013122  006302                            ASL     R2
4165  013124  006302                            ASL     R2
4166  013126  000302                            SWAB    R2
4167  013130  042702  177760                    BIC     #177760,R2      ;STRIP AWAY ALL JUNK
4168  013134  004767  007536                    JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4169  013140  004567  007752                    JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
4170  013144  025144                            LINE
4171  013146  026600                            EM20+36
4172  013150  104020                            ERROR   20              ;RECEIVER SPEED INCORRECT
4173
4174  013152  016767  012016  012016  8$:       MOV     TIMEB,TIMEC     ;SET UP NEW CHECK TIMER COUNT
4175  013160  062767  002100  166030            ADD     #2100,$TMP7     ;GENERATE NEXT SPEED
4176  013166  022767  035600  166022            CMP     #35600,$TMP7    ;DONE ALL SPEEDS ?
4177  013174  001255                            BNE     2$              ;BR IF NOT
4178  013176  000643                            BR      1$              ;GO TEST NEXT LINE
4179
```

```
4182                                    ;*****************************************************************
 (3)                                    ;#TEST 43       BASIC DATA TEST - ALL LINES/ALL CHAR LENGTHS
 (3)                                    ;*****************************************************************
 (2)  013200  000004                    TST43:  SCOPE
 (1)  013202  012767  000043  166024             MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
4183                                    .REM    X
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST VERIFIES THAT A SINGLE ALL ONES CHAR. CAN BE TRANS-
 (1)                                    MITTED AND RECEIVED ON ALL SELECTED LINES AT ALL FOUR CHAR LENGTHS
 (1)                                    (5, 6, 7, AND 8 BITS). THE TEST SEQUENCE IS AS FOLLOWS:
 (1)
 (1)                                            1. SET UP THE ERROR LOOP RETURN
 (1)                                            2. SELECT A LINE NO. TO TEST - GO TO TEST 43 IF DONE ALL
 (1)                                               SELECTED LINES.
 (1)                                            3. GET A TEST CHARACTER FROM THE DATA TABLE AND UPDATE THE
 (1)                                               TABLE POINTER.
 (1)                                            4. CLEAR THE DH11
 (1)                                            5. PRIME THE SELECTED LINE TO XMIT ONE CHAR AT 9600 BAUD
 (1)                                            6. WAIT FOR "CHAR AVAIL" TO SET - IF TIMEOUT REPORT ERROR AND
 (1)                                               RESTART AT STEP 8.
 (1)                                            7. IF NO TIMEOUT - CHECK DATA AND REPORT ANY ERRORS
 (1)                                            8. INCREMENT "SCR" REG TO CHANGE CHAR LENGTH - IF DONE
 (1)                                               ALL FOUR GO TO STEP 2 - IF NOT THEN STEP 4.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1. [ERROR  55] IS CALLED TO REPORT RCVR TIMEOUT.
 (1)                                    2. [ERROR  23] IS CALLED TO REPORT DATA COMPARE ERRORS
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    ******
 (1)
 (1)                                    1. IF FAULT AFFECTS ONLY ONE LINE AT ALL CHAR LENGTHS, SUSPECT A
 (1)                                       BAD UART MODULE M7280.
 (1)
 (1)                                    2. IF FAULT AFFECTS ONLY ONE BIT ON ALL LINES, SUSPECT THE
 (1)                                        THE M7279 MODULE.
 (1)                                    3. IF FAULT AFFECTS ONLY CERTAIN CHAR LENGTHS, SUSPECT EITHER THE M7278
 (1)                                       OR THE UART MODULE M7280.
 (1)
 (1)                                    KEY LOGIC:
 (1)                                    **********
 (1)
 (1)                                            M7280'S UART CHIPS PINS <12:05>
 (1)
 (1)                                            M7279   SH1     E1,E2,E6, OR E7
 (1)
 (1)                                            M7278   SH8     N82 LPR 01 H    FH1
```

```
  (1)                                                    NB1 LPR 00 H    FH?
  (1)                                  %
 4184  013210  012767  013236  165672          MOV     #3$,$LPERR      ;SET UP ERROR LOOP RETURN
 4185  013216  004767  007602                1$: JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
 4186  013222  000511                          BR      TST44           ;BR IF DONE ALL SELECTED LINES
 4187  013224  012705  025160                  MOV     #TDATA2,R5      ;GET POINTER TO DATA TABLE
 4188  013230  005002                          CLR     R2              ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
 4189  013232  012567  165760                2$: MOV     (R5)+,$TMP7     ;PUT TEST CHAR IN XMIT BUFFER
 4190  013236  012711  004000                3$: MOV     #BIT11,(R1)     ;CLEAR THE DH11
 4191  013242  156711  011676                  BISB    LINE,(R1)       ;SELECT THE LINE
 4192  013246  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET BYTE COUNT TO -1
 4193  013254  012761  001216  000006          MOV     #$TMP7,CAR(R1)  ;SET CURRENT ADDRESS REG
 4194  013262  012761  033500  000004          MOV     #33500,LPR(R1)  ;SET BAUD RATE TO 9600
 4195  013270  050261  000004                  BIS     R2,LPR(R1)      ;SELECT CHAR LENGTH
 4196  013274  156761  011112  000012          BISB    LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4197
 4198  013302  012767  000001  011662          MOV     #1,TIMEA        ;INIT TIMER A
 4199  013310  005067  011660                  CLR     TIMEB           ;INIT TIMER B
 4200  013314  105711                        4$: TSTB    (R1)            ;RCVR DONE YET ??
 4201  013316  100424                          BMI     5$              ;BR IF YES
 4202  013320  004767  010626                  JSR     PC,TIMEIT       ;CALL THE TIMER
 4203  013324  000773                          BR      4$              ;TIMER ROUTINE WILL MOVE RETURN PC
 4204                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4205
 4206  013326  004767  011002                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4207  013332  011103                          MOV     (R1),R3         ;GET THE SCR
 4208  013334  042703  177560                  BIC     #177560,R3      ;CLEAR UNINTERESTING BITS
 4209  013340  012704  000200                  MOV     #200,R4         ;SET UP S/B DATA
 4210  013344  156704  011574                  BISB    LINE,R4
 4211  013350  004767  007322                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4212  013354  004567  007536                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
 4213  013360  025144                          LINE
 4214  013362  026720                          EM22+51
 4215  013364  104055                          ERROR   55              ;CHAR AVAIL FAILED TO SET ON TIME
 4216  013366  000422                          BR      6$              ;GO TEST NEXT CHAR LENGTH
 4217
 4218  013370  016103  000002              55: MOV     NRC(R1),R3      ;GET THE WAS DATA
 4219  013374  012704  000200                  MOV     #200,R4         ;SET UP THE S/B DATA IN R4
 4220  013400  156704  011540                  BISB    LINE,R4
 4221  013404  000304                          SWAB    R4
 4222  013406  156704  165604                  BISB    $TMP7,R4
 4223  013412  020304                          CMP     R3,R4           ;WAS THE RCVD DATA CORRECT ??
 4224  013414  001407                          BEQ     6$              ;BR IF YES
 4225
 4226  013416  004767  007250                  JSR     PC,SUER2        ;GO SET UP THE ERROR INFO
 4227  013422  004567  007470                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG
 4228  013426  025144                          LINE
 4229  013430  026761                          EM23+36
 4230  013432  104023                          ERROR   23              ;DATA COMPARE ERROR
 4231
 4232  013434  005202                        6$: INC     R2              ;DO NEXT CHAR LENGTH ON SELECTED LINE
 4233  013436  022702  000004                  CMP     #4,R2           ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
 4234  013442  001273                          BNE     2$              ;BR IF NOT
 4235  013444  000664                          BR      1$              ;GO DO NEXT LINE
```

```
4238                                    ;*****************************************************************
 (3)                                    ;*TEST 44       SINGLE LINE DATA TEST - ALL LINES
 (3)                                    ;*****************************************************************
 (2)  013446  000004                    TST44:  SCOPE
 (1)  013450  012767  000044  165556            MOV     #STN-1,STESTN   ;;SET TEST NUMBER IN  MAIL BOX
4239                                    .REM    %
 (1)                                    TEST ABSTRACT:
 (1)                                    **************
 (1)
 (1)                                            THIS TEST TRANSMITS AND RECEIVES A BINARY COUNT PATTERN
 (1)                                    (000 - 377) ON ALL SELECTED LINES. THE TEST SEQUENCE IS AS
 (1)                                    FOLLOWS:
 (1)
 (1)                                            1. SET UP THE ERROR LOOP RETURN
 (1)                                            2. GO SELECT A LINE NO. TO TEST - IF DONE ALL SELECTED
 (1)                                               LINES THEN GO TO TEST 44.
 (1)                                            3. CLEAR THE DH11 AND PRIME THE SELECTED LINE TO XMIT
 (1)                                               TO XMIT 256. CHARS AT 9600. BAUD - 8 BIT CHARS.
 (1)                                            4. SET UP R5 TO POINT TO RCVR CORE BUFFER.
 (1)                                            5. ACTIVATE THE SELECTED XMITTER.
 (1)                                            6. WAIT FOR "CHAR AVAIL" TO SET BEFORE READING THE SILO.
 (1)                                               IF RCVR TIMEOUT REPORT ERROR AND RESTART AT STEP 2.
 (1)                                            7. IF NO TIMEOUT READ THE SILO AND STORE THE WORD IN THE RCVR
 (1)                                               CORE BUFFER - WHEN THE BUFFER IS FULL GO TO STEP8 IF NOT
 (1)                                               THEN GO TO STEP 6.
 (1)                                            8. COMPARE THE XMIT AND RCVR CORE IMAGE BUFFERS AND REPORT
 (1)                                               ALL DATA COMPARE ERRORS.
 (1)                                            9. CHECK THE "BAR", "BCR" AND "CAR" REGISTERS FOR CORRECT
 (1)                                               CONTENTS - REPORT ALL ERRORS.
 (1)                                            10.GO TO STEP 2
 (1)
 (1)                                                    NOTE:   THIS TEST USES THE MAINT. BIT (SCRD9=1) TO
 (1)                                                            TURN THE DATA AROUND INSTEAD OF THE MAINT
 (1)                                                            CONNECTORS OR MODULE.
 (1)
 (1)                                    ERRORS:
 (1)                                    *******
 (1)
 (1)                                    1. [ERROR  22] IS CALLED TO REPORT "DATA AVAIL" TIMEOUT
 (1)                                    2. [ERROR  37]  "        "          " DATA COMPARE ERRORS
 (1)                                    3. [ERROR  40]  "        "          " "BAR" REG NOT CLEARED
 (1)                                    4. [ERROR  10]  "        "          " "BCR" REG NOT ALL ZEROES
 (1)                                    5. [ERROR   7]  "        "          " "CAR" REG NOT UPDATED CORRECTLY
 (1)
 (1)                                    SYNC:   M7277   SH3     INIT A H        EF2
 (1)                                    *****
 (1)
 (1)                                    DEBUG:
 (1)                                    \******
 (1)
 (1)                                    1. IF THE FAULT AFFECTS ONE OR MORE LINES IN AN 8 LINE GROUP <15:08>
 (1)                                       OR <07:00>, SWAP THE M7280 MODULES. IF THE FAULT SHIFTS SO THAT
 (1)                                       THE ERROR INDICATES DIFFERENT LINES THE PROBLEM IS MOST LIKELY
 (1)                                       THE M7280 THE SYMPTOM SHIFTED TO.
```

```
(1)
(1)                                     2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON
(1)                                        ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.
(1)
(1)                                     3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS
(1)                                        ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.
(1)
(1)                                     4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS
(1)                                        SUSPECT THE M7278 OR M7277 MODULES RESPECTFULLY.
(1)
(1)                                     KEY LOGIC       (REFER TO TEST 43)
(1)                                     ***********
(1)
(1)                                     %
4240  013456  012767  013500  165424          MOV     #2$,SLPERR      ;SET UP ERROR LOOP RETURN
4241  013464  004767  007334            1$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4242  013470  000401                           BR      11$             ;;BR IF ALL SELECTED LINES DONE
4243  013472  000402                           BR      2$              ;GO TEST IT
4244  013474  000167  000424           11$:    JMP     @$              ;EXIT TEST
4245  013500  016701  010674            2$:    MOV     DHADR,R1        ;RESET DEVADR
4246  013504  012711  004000                   MOV     #BIT11,(R1)     ;CLEAR THE DH11
4247  013510  156711  011430                   BISB    LINE,(R1)       ;SET SELECT BITS IN SCR
4248  013514  012761  033436  000006           MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG
4249  013522  012761  177400  000010           MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT
4250  013530  012761  033503  000004           MOV     #33503,LPR(R1)  ;SET LINE PARAMETERS
4251  013536  012705  032436                   MOV     #RBUF,R5        ;SET UP POINTER TO INPUT DATA BUFFER
4252  013542  052711  001000                   BIS     #BIT09,(R1)     ;SET MAINT MODE BIT
4253  013546  016761  010640  000012           MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
4254
4255  013554  012767  000002  011410           MOV     #2,TIMEA        ;INIT TIMER A
4256  013562  005067  011406                   CLR     TIMEB           ;INIT TIMER B
4257  013566  105711                   3$:     TSTB    (R1)            ;RCVR DONE YET ??
4258  013570  100425                           BMI     4$              ;BR IF YES
4259  013572  004767  010354                   JSR     PC,TIMEIT       ;CALL THE TIMER
4260  013576  000773                           BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4261                                                                   ;AROUND THIS BRANCH IF TIME OUT OCCURS
4262
4263  013600  004767  010530                   JSR     PC,SAPS         ;SAVE THE ERROR PSW
4264  013604  010102                           MOV     R1,R2           ;SET UP REGADR
4265  013606  011203                           MOV     (R2),R3         ;GET THE WAS DATA
4266  013610  042703  176400                   BIC     #176400,R3      ;CLEAR JUNK BITS
4267  013614  012704  001200                   MOV     #1200,R4        ;SET UP S/B DATA
4268  013620  156704  011320                   BISB    LINE,R4
4269  013624  004767  007046                   JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4270  013630  004567  007262                   JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4271  013634  025144                           LINE
4272  013636  026720                           EM22+51
4273  013640  104022                           ERROR   22              ;CHAR AVAIL TIMEOUT
4274  013642  000710                           BR      1$              ;GO TRY NEXT LINE
4275
4276  013644  016125  000002           4$:     MOV     NRC(R1),(R5)+   ;SAVE THE RECEIVED DATA
4277  013650  022705  033436                   CMP     #RBUF+1000,R5   ;INPUT BUFFER FULL ??
4278  013654  001344                           BNE     3$              ;BR IF NOT
4279
```

```
4280  013656  012702  033436                MOV     #TBUF,R2        ;SET UP POINTER TO OUTPUT BUFFER
4281  013662  012701  032436                MOV     #RBUF,R1        ;SET UP POINTER TO INPUT BUFFER
4282  013666  111204                5$:     MOVB    (R2),R4         ;SET UP S/B DATA IN R4
4283  013670  042704  177400                BIC     #177400,R4
4284  013674  000304                        SWAB    R4
4285  013676  156704  011242                BISB    LINE,R4
4286  013702  152704  000200                BISB    #200,R4
4287  013706  000304                        SWAB    R4
4288  013710  011103                        MOV     (R1),R3         ;GET THE WAS DATA
4289  013712  020304                        CMP     R3,R4           ;DATA CORRECT ??
4290  013714  001407                        BEQ     6$              ;BR IF YES
4291
4292  013716  004767  006750                JSR     PC,SUER2        ;GO SET UP ERROR INFO
4293  013722  004567  007170                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4294  013726  025144                        LINE
4295  013730  030176                        EM37+33
4296  013732  104037                        ERROR   37              ;DATA COMPARE ERROR
4297
4298
4299  013734  005202                6$:     INC     R2              ;UPDATE DATA BUFFER POINTERS
4300  013736  062701  000002                ADD     #2,R1           ;
4301  013742  022701  033436                CMP     #RBUF+1000,R1   ;COMPARED ALL 256. CHARS ??
4302  013746  001347                        BNE     5$              ;BR IF NOT
4303
4304  013750  016701  010424                MOV     DHADR,R1        ;RESET DEVADR
4305  013754  010102                        MOV     R1,R2           ;SET UP REGADR
4306  013756  062702  000012                ADD     #BAR,R2
4307  013762  005712                        TST     (R2)            ;WAS THE "BAR" ALL ZEROES ??
4308  013764  001413                        BEQ     7$              ;BR IF YES
4309
4310  013766  004767  010342                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4311  013772  011203                        MOV     (R2),R3         ;GET THE WAS DATA
4312  013774  005004                        CLR     R4              ;SET UP S/B DATA
4313  013776  004767  006674                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4314  014002  004567  007110                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4315  014006  025144                        LINE
4316  014010  030241                        EM40+40
4317  014012  104040                        ERROR   40              ;"BAR" REG NOT ALL ZEROES
4318
4319  014014  010102                7$:     MOV     R1,R2           ;SET UP REGADR
4320  014016  062702  000010                ADD     #BCR,R2
4321  014022  005712                        TST     (R2)            ;BYTE COUNT REG ALL ZEROES ?
4322  014024  001413                        BEQ     71$             ;;BR IF BYTE COUNT ZERO
4323
4324  014026  004767  010302                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4325  014032  011203                        MOV     (R2),R3         ;GET THE WAS DATA
4326  014034  005004                        CLR     R4              ;SET UP THE S/B DATA
4327  014036  004767  006634                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4328  014042  004567  007050                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4329  014046  025144                        LINE
4330  014050  026046                        EM10+44
4331  014052  104010                        ERROR   10              ;BYTE COUNT NOT ALL ZEROES
4332
4333  014054  010102                71$:    MOV     R1,R2           ;SET UP REGADR
```

```
 4334  014056  062702  000006                ADD     #CAR,R2
 4335  014062  022712  034036                CMP     #TBUF+400,(R2)  ;DID "CAR" INCREMENT PROPERLY ?
 4336  014066  001414                        BEQ     72$             ;BR IF YES
 4337
 4338  014070  004767  010240                JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4339  014074  011203                        MOV     (R2),R3         ;GET THE WAS DATA
 4340  014076  012704  034036                MOV     #TBUF+400,R4    ;SET UP S/B DATA
 4341  014102  004767  006570                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4342  014106  004567  007004                JSR     R5,SUNUM        ;GO PUT LINE NO IN MESSAGE
 4343  014112  025144                        LINE
 4344  014114  025777                        EM7+47
 4345  014116  104007                        ERROR   7               ;"CAR" NOT UPDATED CORRECTLY
 4346
 4347  014120  000167  177340        72$:    JMP     1$              ;GO DO NEXT LINE
 4348
 4349  014124  000240                 8$:    NOP                     ;EXIT POINT
```

```
                                        ;*****************************************************************
                                        ;*TEST 45       BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY
                                        ;*****************************************************************
014126  000004                          TST45:  SCOPE
014130  012767  000045  165076                  MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                        .REM    %
                                        TEST ABSTRACT:
                                        ***************

                                                THIS TEST VERIFIES THE ODD PARITY FUNCTION FOR ALL SELECTED
                                        LINES USING THE "BREAK" FUNCTION TO FORCE PARITY ERRORS. REFER TO
                                        THE FLOW CHARTS IN THE PROGRAM DOCUMENTATION FOR TEST SEQUENCES.

                                        ERRORS:
                                        *******

                                        1. [ERROR  22] IS CALLED TO REPORT RCVR TIMEOUT
                                        2. [ERROR  33] IS CALLED TO REPORT DATA/PARITY ERRORS

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1. IF FAULT AFFECTS ALL LINES SUSPECT THE M7278 MODULE.

                                        2. IF IT AFFECTS ONLY ONE LINE SUSPECT THE "UART" MODULE FOR THAT LINE.


                                        KEY LOGIC:
                                        **********

                                                M7278   SH7     PEN LPR04 L     FF2
                                                                PEV LPR05 L     FN1
                                        %
014136  012767  011152  164744                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
014144  004767  006654                  1$:     JSR     PC,$ELINE       ;GO SELECT A LINE NO.
014150  000506                          2$:     BR      TST46           ;;BR IF ALL SELECTED LINES DONE
014152  012711  004000                  2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
014156  012704  000200                          MOV     #200,R4         ;SET UP THE S/B DATA IN R4
014162  156704  010756                          BISB    LINE,R4
014166  000304                                  SWAB    R4
014170  156711  010750                          BISB    LINE,(R1)       ;SET LINE NO. IN SCR
014174  012767  000377  010754                  MOV     #377,TDATA1     ;LOAD XMIT BUFFER WITH TEST CHARACTER
014202  012761  073563  000004                  MOV     #73563,LPR(R1)  ;SET UP THE LINE PARAMETERS
014210  012761  177777  000010                  MOV     #-1,BCR(R1)     ;LOAD THE BYTE COUNT REG
014216  012761  025156  000006                  MOV     #TDATA1,CAR(R1) ;LOAD THE BUS ADDR REG
014224  016761  010162  000014                  MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR SELECTED LINE
014232  016761  010154  000012                  MOV     LINMSK,BAR(R1)  ;ACTIVATE THE XMITTR

014240  012767  000001  010724                  MOV     #1,TIMEA        ;INIT TIMER A
014246  005067  010722                          CLR     TIMEB           ;INIT TIMER B
014252  105711                          3$:     TSTB    (R1)            ;RCVR DONE YET ??
```

```
4373  014254  100423                          BMI     4$              ;BR IF YES
4374  014256  004767  007670                  JSR     PC,TIMEIT       ;CALL THE TIMER
4375  014262  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4376                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4377
4378  014264  004767  010044                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4379  014270  011103                          MOV     (R1),R3         ;GET THE WAS DATA
4380  014272  012704  100200                  MOV     #100200,R4      ;SET UP THE S/B DATA
4381  014276  156704  010642                  BISB    LINE,R4
4382  014302  010102                          MOV     R1,R2           ;SET UP REGADR
4383  014304  004767  006366                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4384  014310  004567  006602                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4385  014314  025144                          LINE
4386  014316  026720                          EM22+52
4387  014320  104022                          ERROR   22              ;TIMED OUT WAITING FOR DATA AVAIL
4388  014322  000710                          BR      1$              ;GO TEST NEXT LINE
4389
4390  014324  016103  000002          4$:     MOV     NRC(R1),R3      ;GET THE WAS DATA
4391  014330  020304                          CMP     R3,R4           ;CORRECT DATA RECEIVED ??
4392  014332  001704                          BEQ     1$              ;BR IF YES
4393
4394  014334  004767  007774                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4395  014340  010102                          MOV     R1,R2           ;SET UP THE REGADR
4396  014342  062702  000002                  ADD     #NRC,R2
4397  014346  004767  006324                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4398  014352  004567  006540                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4399  014356  025144                          LINE
4400  014360  027712                          EM33+40
4401  014362  104033                          ERROR   33              ;INCORRECT DATA OR PARITY ERROR
4402  014364  000667                          BR      1$              ;GO TEST NEXT LINE
```

```
                                ;*****************************************************************
                                ;*TEST 46       MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
                                ;*****************************************************************
014366  000004                  TST46:  SCOPE
014370  012767  000046  164636          MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    X
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE
                                A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7,
                                AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED.
                                THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH
                                SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB:".  REFER TO THE FLOW
                                CHARTS IN THE PROGRAM DOCUMENT FOR A DETAILED DESCRIPTION OF THE TEST
                                SEQUENCES.

                                ERRORS:
                                *******

                                1. [ERROR  41] IS CALLED TO REPORT FALSE RECEIVER INTRS.
                                2. [ERROR  42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
                                3. [ERROR  34] IS CALLED TO REPORT PARITY/DATA ERRORS
                                4. [ERROR  35] IS CALLED TO REPORT TEST TIMEOUTS

                                SYNC:   (NONE)
                                *****

                                DEBUG:  (REFER TO TEST 45)
                                ******

                                KEY LOGIC:      (REFER TO TEST 45)
                                **********

                                %
014376  012767  014414  164504           MOV     #1$,SLPERR      ;SET UP THE ERROR LOOP RETURN
014404  012705  024462                   MOV     #PRTYTB+4,R5    ;SET UP POINTER TO TEST PARAMETERS
014410  005067  164602                   CLR     STMP7           ;START WITH SUB TEST #00
014414  162705  000004           1$:     SUB     #4,R5           ;RESET POINTER FOR ERROR LOOPS
014420  005367  164572                   DEC     STMP7           ;RESET SUB TEST # FOR ERROR LOOP
014424  022705  024516           2$:     CMP     #PRTYTB+40,R5   ;DONE ALL 8. SUB TESTS ??
014430  001456                   BEQ     21$             ;BR IF YES
014432  012706  001100                   MOV     #STACK,SP       ;RESET STACK POINTER FOR ERROR LOOPS
014436  016701  007736                   MOV     DHADR,R1        ;RESET DEVADR FOR ERROR LOOPS
014442  012567  164546                   MOV     (R5)+,STMP6     ;GET THE BYTE COUNT PARAMETER
014446  012567  164540                   MOV     (R5)+,STMP5     ;GET THE LINE PARAMETERS
014452  005267  164540                   INC     STMP7           ;GENERATE NEW SUB-TEST NO.
014456  012711  004000                   MOV     #BIT11,(R1)     ;CLEAR THE DH11
014462  004767  006612                   JSR     PC,SUPPAR       ;GO SET UP PARAMETERS
014466  016167  000004  164504           MOV     LPR(R1),STMP0   ;SAVE CURRENT LINE PARAMETERS
014474  016767  007710  164504           MOV     LINSEL,STMP3    ;SAVE SELECTED LINES PARAMETER
014502  004767  007612                   JSR     PC,CHPS2        ;GO LOCK OUT INTRS
014506  016702  007670                   MOV     DHVCT,R2        ;SET UP THE VECTOR
014512  012722  014604                   MOV     #3$,(R2)+       ;GO TO 3$ ON RCVR INTERRUPT
```

```
4426  014516  116712  010416                    MOVB    DHRLVL,(R2)
4427  014522  012711  000100                    MOV     #100,(R1)       ;ENABLE CHAR AVAIL INTERRUPTS
4428  014526  016767  007656  010122            MOV     LINSEL,LINACT   ;FLAG ALL SELECTED LINES ACTIVE
4429  014534  016761  007650  000012            MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTEDLINES
4430  014542  116767  164334  164434            MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
4431  014550  042767  177400  164426            BIC     #177400,$TMP2
4432  014556  004767  007522                    JSR     PC,CHPS1        ;GO CLEAR PSW
4433  014562  000167  000176                    JMP     7$              ;GO WAIT FOR INTERRUPTS
4434
4435  014566  012706  001100          21$:      MOV     #STACK,SP       ;RESTORE THE SP
4436  014572  004767  007506                    JSR     PC,CHPS1        ;GO CLEAR PSW
4437  014576  004767  007322                    JSR     PC,RESTRP       ;RESTORE TRAP CATCHER
4438  014602  000536                            BR      TST47           ;;GO TO NEXT TEST
4439
4440                                  ;RECEIVER INTERRUPT SERVICE ROUTINE
4441
4442  014604  105711                  3$:       TSTB    (R1)            ;CHAR AVAIL SET
4443  014606  100404                            BMI     4$              ;BR IF YES
4444
4445  014610  012711  004000                    MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4446  014614  104041                            ERROR   41              ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
4447  014616  000702                            BR      2$              ;GO TRY NEXT SUB TEST
4448
4449  014620  032711  040000          4$:       BIT     #BIT14,(R1)     ;SILO OVERFLOW ??
4450  014624  001404                            BEQ     5$              ;BR IF NOT
4451
4452  014626  012711  004000                    MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4453  014632  104042                            ERROR   42              ;SILO OVERFLOW ERROR
4454  014634  000673                            BR      2$              ;GO TRY NEXT SUB TEST
4455
4456  014636  016103  000002          5$:       MOV     RBUF(R1),R3     ;GET THE WAS DATA
4457  014642  010302                            MOV     R3,R2           ;EXTRACT AND SAVE LINE NO.
4458  014644  000302                            SWAB    R2
4459  014646  042702  177760                    BIC     #177760,R2
4460  014652  010267  164332                    MOV     R2,$TMP4
4461  014656  006302                            ASL     R2              ;GENERATE TABLE OFFSETR
4462  014660  026203  032436                    CMP     RBUF(R2),R3     ;CORRECT DATA RECEIVED ??
4463  014664  001426                            BEQ     6$              ;BR IF YES
4464
4465  014666  004767  007442                    JSR     PC,SAPS         ;SAVE THE ERROR PSW
4466  014672  012711  004000                    MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4467  014676  016204  032436                    MOV     RBUF(R2),R4     ;SET UP S/B DATA
4468  014702  062701  000002                    ADD     #2,R1           ;SET UP WAS ADDRESS
4469  014706  062702  032436                    ADD     #RBUF,R2        ;SET UP S/B ADDRESS
4470  014712  004767  005760                    JSR     PC,SETUP2A      ;GO SET UP ERROR INFO
4471  014716  004567  006174                    JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4472  014722  001210                            $TMP4
4473  014724  027766                            EM34+51
4474  014726  004567  006164                    JSR     R5,SUNUM        ;PUT SUBTEST NO. IN MESSAGE
4475  014732  001216                            $TMP7
4476  014734  030004                            EM34+67
4477  014736  104034                            ERROR   34              ;PARITY DATA COMPARE ERROR
4478  014740  000631                            BR      2$              ;GO TRY NEXT SUBTEST
4479
```

```
4480  014742  105262  032436          6$:     INCB    RBUF(R2)        ;GENERATE NEW RCVD DATA
4481  014746  005262  024660                  INC     MULPTB(R2)      ;COUNT ONE BYTES RECEIVED
4482  014752  001003                          BNE     61$             ;BR IF NOT DONE
4483  014754  046267  024616  007674          BIC     LINBIT(R2),LINACT ;FLAG THIS LINE DONE
4484  014762  000002                  61$:    RTI                     ;RETURN TO WAIT ROUTINE
4485
4486                                  ;WAIT ROUTINE
4487
4488  014764  012767  000002  010200  7$:     MOV     #2,TIMEA        ;INIT TIMER A
4489  014772  005067  010176                  CLR     TIMEB           ;INIT TIMER B
4490  014776  005761  000012          8$:     TST     BAR(R1)         ;ALL LINES DONE XMITTING ??
4491  015002  001413                          BEQ     9$              ;BR IF YES
4492  015004  004767  007142                  JSR     PC,TIMEIT       ;CALL THE TIMER
4493  015010  000772                          BR      8$              ;TIMER ROUTINE WILL MOVE RETURN PC
4494                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4495
4496  015012  016167  000012  164166          MOV     BAR(R1),$TMP3   ;SAVE THE ACTIVE LINES FLAG
4497  015020  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4498  015024  104035                          ERROR   35              ;TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
4499  015026  000167  177372                  JMP     2$              ;GO TRY NEXT SUBTEST
4500
4501
4502  015032  012767  000001  010132  9$:     MOV     #1,TIMEA        ;INIT TIMER A
4503  015040  005067  010130                  CLR     TIMEB           ;INIT TIMER B
4504  015044  005767  007606          10$:    TST     LINACT          ;ALL CHARS RECEIVED ?
4505  015050  001411                          BEQ     11$             ;BR IF YES
4506  015052  004767  007074                  JSR     PC,TIMEIT       ;CALL THE TIMER
4507  015056  000772                          BR      10$             ;TIMER ROUTINE WILL MOVE RETURN PC
4508                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4509
4510  015060  016767  007572  164120          MOV     LINACT,$TMP3    ;SET UP ACTIVE LINE PARAMETER
4511  015066  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
4512  015072  104035                          ERROR   35              ;SILO EMPTY TIMEOUT
4513  015074  000167  177324          11$:    JMP     2$              ;GO TRY NEXT SUB TEST
```

```
4515                                ;*****************************************************************
 (3)                                ;*TEST 47       AUTO ECHO TEST 1 - ALL LINES
 (3)                                ;*****************************************************************
 (2)  015100  000004                TST47:  SCOPE
 (1)  015102  012767  000047  164124        MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
4516                                .REM    X
 (1)                                TEST ABSTRACT:
 (1)                                **************
 (1)
 (1)                                        THIS TEST VERIFIES THAT ALL SELECTED LINES CAN TURN AROUND
 (1)                                A SINGLE TEST CHARACTER (377) AND (000) IN AUTO ECHO MODE. THE TEST SEQUENCE
 (1)                                IS AS FOLLOWS:
 (1)
 (1)                                        1. SET UP THE ERROR LOOP RETURN
 (1)                                        2. RETRIEVE THE AUTO-ECHO TEST DATA FROM "AETAB:" AND
 (1)                                           UPDATE THE POINTER.
 (1)                                        3. GO SELECT A LINE NO. TO TEST - GO TO STEP 10 IF DONE
 (1)                                           ALL SELECTED LINES.
 (1)                                        4. CLEAR THE "CAR" AND "BCR" MEMORIES.
 (1)                                        5. PRIME THE SELECTED LINE TO XMIT ONE CHAR WITH A.E. ENABLED.
 (1)                                        6. ACTIVATE THE SELECTED TRANSMITTER.
 (1)                                        7. WAIT FOR "CHAR AVAIL" - IF TIMEOUT REPORT ERROR AND RESTART
 (1)                                           AT STEP 2.
 (1)                                        8. IF NO TIMEOUT - READ SILO AND COMPARE AUTO ECHO DATA RECEIVED
 (1)                                           REPORT DATA COMPARE ERRORS AND RESTART AT STEP 2
 (1)                                        9. IF NO ERRORS REPEAT STEPS 7 AND 8 SIXTY-FOUR TIMES THEN TURN
 (1)                                           OFF A.E. ENABLE AND READ LAST CHAR FROM SILO.. CHECK LAST
 (1)                                           CHAR FOR DATA COMPARE ERRORS - REPORT ERRORS IF ANY - AND RE-
 (1)                                           START AT STEP 2.
 (1)                                       10. CHANGE A.E. TABLE POINTER TO POINT TO "AETAB0:" (0'S DATA)
 (1)                                           AND REPEAT STEPS 2 THRU 9.
 (1)
 (1)                                ERRORS:
 (1)                                ********
 (1)
 (1)                                1. [ERROR 24] IS CALLED TO REPORT ALL ERRORS
 (1)
 (1)                                SYNC:   M7277   SH4     LOAD BAR LB+1B L        CN2
 (1)                                *****
 (1)
 (1)                                DEBUG:
 (1)                                ******
 (1)
 (1)                                1. IF ALL LINES FAIL, SUSPECT EITHER THE M7277 OR M7289
 (1)
 (1)                                2. IF ONLY ONE LINE FAILS SUSPECT THE M7288
 (1)
 (1)                                3. LOOP ON THE FAILING LINE AND TRACK BACK THROUGH THE KEY LOGIC.
 (1)
 (1)                                KEY LOGIC:
 (1)                                ***********
 (1)
 (1)                                        M7277   SH3     AE GO L         CS2
```

```
(1)                                                     7402 "OR" GATE CHIPS E38 OR E41
(1)                                                     74157 MUX CHIPS  E39 OR 342
(1)                                             SH4     E35 - PIN 2 STUCK LOW
(1)
(1)                                     M7289   SH3     AE GO L          EK1
(1)                                                     AE SCAN MUX  E22 PIN 10
(1)                                             SH4     SAMPLE STATUS H E21-12
(1)
(1)                                     M7288   SH5, 7, 9, 11
(1)                                                     AE ENABLE "NN" H CONTROL FLOPS
(1)                                                     74174 CHIPS PIN 15
(1)                                     %
4517  015110  012767  015172  163772          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4518  015116  005067  164064                  CLR     $TMP3           ;INIT I/O DATA FLAG
4519  015122  012711  004000                  MOV     #BIT11,(R1)     ;CLEAR THE DH11
4520  015126  012705  024516                  MOV     #AETAB,R5       ;GET POINTER TO AUTO ECHO DATA TABLE
4521  015132  005067  007256          7$:     CLR     LMSK1           ;INIT BIT TEST MARKER
4522  015136  000261                          SEC                     ;SET "C" BIT FOR MARKER
4523  015140  000401                          BR      13$             ;GO SHIFT MASK
4524  015142  000241                  1$:     CLC                     ;INIT THE "C" BIT
4525  015144  006167  007244          13$:    ROL     LMSK1           ;SHIFT BIT MARKER
4526  015150  001407                          BEQ     12$             ;BR IF DONE ALL LINES
4527  015152  012504                          MOV     (R5)+,R4        ;SET UP THE S/B DATA
4528  015154  033767  007234  007226          BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
4529  015162  001767                          BEQ     1$              ;BR IF NOT
4530  015164  004767  005634          11$:    JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4531  015170                          12$:
(1)   015170  000522                          BR      6$              ;;BR IF ALL SELECTED LINES TESTED
4532  015172  004767  006000          2$:     JSR     PC,CLCABC       ;GO CLEAR "CAR" AND "BCR" MEMORIES
4533  015176  116711  007742                  MOVB    LINE,(R1)       ;SET SELECT BITS IN SCR REG
4534  015202  012761  177777  000010          MOV     #-1,BCR(R1)     ;SET UP TO XFER ONE CHAR
4535  015210  010561  000006                  MOV     R5,CAR(R1)      ;SET UP THE BUS ADDRESS REG
4536  015214  162761  000002  000006          SUB     #2,CAR(R1)      ;CORRECT BUS ADDRESS
4537  015222  012767  000100  163766          MOV     #100,$TMP7      ;COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
4538  015230  005067  163760                  CLR     $TMP6           ;INIT CHAR COUNTER
4539  015234  012761  133503  000004          MOV     #133503,LPR(R1) ;SET UP LINE PARAMETER REG
4540  015242  016761  007144  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE THE LINE
4541
4542  015250  012767  000002  007714          MOV     #2,TIMEA        ;INIT TIMER A
4543  015256  005067  007712                  CLR     TIMEB           ;INIT TIMER B
4544  015262  105711                  3$:     TSTB    (R1)            ;CHAR AVAIL SET ??
4545  015264  100427                          BMI     4$              ;BR IF YES
4546  015266  004767  006660                  JSR     PC,TIMEIT       ;CALL THE TIMER
4547  015272  000773                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4548                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4549
4550  015274  004767  007034                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4551  015300  005061  000004                  CLR     LPR(R1)         ;TURN OFF AUTO ECHO MODE
4552  015304  010102                          MOV     R1,R2           ;MAKE REGADR = DEVADR
4553  015306  011103                          MOV     (R1),R3         ;GET THE WAS DATA
4554  015310  042703  100000                  BIC     #BIT15,R3       ;CLEAR JUNK BIT
4555  015314  012704  000200                  MOV     #200,R4         ;SET UP S/B DATA
4556  015320  156704  007620                  BISB    LINE,R4
```

```
4557  015324  004767  005346                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4558  015330  004567  005562                  JSR     R5,SUNUM        ;GO SET LINE NO. IN MSG
4559  015334  025144                          LINE
4560  015336  027116                          EM24+35
4561  015340  104024                          ERROR   24              ;DATA AVAIL FAILED TO SET ON TIME
4562  015342  000677                          BR      1$              ;GO TRY NEXT LINE
4563
4564  015344  005267  163644          4$:     INC     $TMP6           ;COUNT ONE CHAR RECVD
4565  015350  016103  000002                  MOV     NRC(R1),R3      ;GET THE WAS DATA
4566  015354  020304                          CMP     R3,R4           ;WAS CHAR AUTO ECHOED CORRECTLY ?
4567  015356  001417                          BEQ     5$              ;BR IF YES
4568
4569  015360  004767  006750                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4570  015364  005061  000004                  CLR     LPR(R1)         ;DISABLE AUTO ECHO
4571  015370  010102                          MOV     R1,R2           ;SET UP REGADR
4572  015372  062702  000002                  ADD     #NRC,R2
4573  015376  004767  005274                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4574  015402  004567  005510                  JSR     R5,SUNUM        ;PUT LINE NO. IN ERROR MSG
4575  015406  025144                          LINE
4576  015410  027116                          EM24+35
4577  015412  104024                          ERROR   24              ;CHAR AUTO ECHOED INCORRECTLY
4578  015414  000652                          BR      1$              ;GO TRY NEXT LINE
4579
4580  015416  005367  163574          5$:     DEC     $TMP7           ;COUNT ONE CHAR READ OUT OF 64
4581  015422  003317                          BGT     3$              ;BR IF NOT LAST ONE
4582  015424  100646                          BMI     1$              ;;BR IF LAST ONE READ
4583  015426  042761  100000  000004          BIC     #BIT15,LPR(R1)  ;DISABLE AUTO ECHO
4584  015434  000712                          BR      3$              ;GO READ LAST CHAR
4585
4586  015436  005167  163544          6$:     COM     $TMP3           ;TOGGLE I/O FLAG
4587  015442  001406                          BEQ     TST50           ;;BR IF DONE BOTH I/O DATA
4588  015444  005067  007474                  CLR     LINE            ;INIT LINE NO TO 00
4589  015450  012705  024556                  MOV     #AETAB0,R5      ;SET POINTER TO 0'S TABLE
4590  015454  000167  177452                  JMP     7$              ;REPEAT TEST FOR ZERO PATTERNS
4591
```

```
 4594                                   ;*****************************************************************
  (3)                                   ;*TEST 50       AUTO ECHO TEST 2 - ALL LINES
  (3)                                   ;*****************************************************************
  (2)  015460  000004                   TSTSO:  SCOPE
  (1)  015462  012767  000050  163544            MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4595                                   .REM    %
  (1)                                   TEST ABSTRACT:
  (1)                                   ***************
  (1)
  (1)                                           THIS TEST IS SIMILAR TO TEST 47 EXCEPT ALL SELECTED LINES OTHER
  (1)                                   THAN THE A.E. TEST LINE ARE ACTIVELY TURNING AROUND A BINARY COUNT
  (1)                                   TEST PATTERN IN NON-AUTO ECHO MODE AND THE A.E. TEST LINE IS TESTED
  (1)                                   FOR ALL 1'S DATA ONLY.
  (1)
  (1)                                   ERRORS:
  (1)                                   ********
  (1)
  (1)                                   1. [ERROR  32] IS CALLED TO REPORT A.E. TEST TIMEOUTS
  (1)                                   2. [ERROR  31] IS CALLED TO REPORT ALL DATA COMPARE ERRORS
  (1)
  (1)                                   SYNC:   M7277   SH4     LOAD BAR LB+HB L        CN2
  (1)                                   *****
  (1)
  (1)                                   DEBUG:
  (1)                                   ******
  (1)
  (1)                                           REFER TO TEST 47
  (1)
  (1)                                   KEY LOGIC:
  (1)                                   **********
  (1)
  (1)                                           REFER TO TEST 47
  (1)
  (1)                                   %
 4596  015470  012767  015544  163412           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 4597  015476  012705  024516                   MOV     #AETAB,R5       ;SET POINTER TO A.E. TEST DATA TABLE
 4598  015502  005067  006706                   CLR     LMSK1           ;INIT BIT TEST MASK
 4599  015506  000261                           SEC                     ;GENERATER MARKER BIT IN "C"
 4600  015510  000401                           BR      12$             ;GO SHIFT MASK
 4601  015512  000241                   1$:     CLC                     ;INIT THE "C" BIT
 4602  015514  006167  006674           12$:    ROL     LMSK1           ;SHIFT TEST BIT
 4603  015520  001410                           BEQ     11$             ;BR IF TESTED ALL LINES
 4604  015522  012567  163470                   MOV     (R5)+,$TMP7     ;GET THE A.E. TEST DATA FOR THIS LINE
 4605  015526  036767  006662  006654           BIT     LMSK1,LINSEL    ;TEST THIS LINE ?
 4606  015534  001766                           BEQ     1$              ;BR IF NOT
 4607  015536  004767  005262                   JSR     PC,SELINE       ;GO SELECT A LINE
 4608  015542                           11$:
  (2)  015542  000571                           BR      TST51           ;;BR IF DONE ALL SELECTED LINES
 4609
 4610  015544  016701  006630           2$:     MOV     DHADR,R1        ;RESET DEVADR IN CASE OF ERROR LOOP
 4611  015550  012711  004000                   MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
 4612  015554  004767  006414                   JSR     PC,SETALL       ;GO SET UP FOR BINARY COUNT XFER ON
 4613                                                                   ;ALL LINES OTHER THAN THE SELECTED ONE
 4614  015560  156711  007360                   BISB    LINE,(R1)       ;SELECT THE LINE FOR A.E. TEST
```

```
4615  015564  010561  000006                    MOV     R5,CAR(R1)      ;SET BUS ADDR TO XMIT TEST CHAR
4616  015570  162761  000002  000006            SUB     #2,CAR(R1)      ;CORRECT THE ADDRESS
4617  015576  012761  177777  00  10            MOV     #-1,BCR(R1)     ;XMIT ONE CHAR ON THIS LINE
4618  015604  012761  133[illegible]03  00  4   MOV     #133503,LPR(R1) ;DO IT AT 9600 BAUD/8 BITS
4619  015612  116767  163[illegible]4  16  4    MOV     $TSTNM,$TMP2    ;SAVE THE TEST NO.
4620  015620  042767  177400  16  [illegible]     BIC     #177400,$TMP2
4621  015626  046767  006560  00  [illegible]     BIC     LINMSK,LINACT   ;MAKE THIS LINE APPEAR INACTIVE
4622  015634  016761  006550  000012              MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL SELECTED TRANSMITTERS
4623
4624  015642  012767  000002  007322  21$:       MOV     #2,TIMEA        ;INIT TIMER A
4625  015650  005067  007[illegible]0              CLR     TIMEB           ;INIT TIMER B
4626  015654  016103  000002          3$:         MOV     NRC(R1),R3      ;GET THE WAS DATA
4627  015660  100414                              BMI     4$              ;BR IF YES
4628  015662  004767  006264                      JSR     PC,TIMEIT       ;CALL THE TIMER
4629  015666  000772                              BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4630                                                                      ;AROUND THIS BRANCH IF TIME OUT OCCURS
4631
4632  015670  016167  000004  163302              MOV     LPR(R[illegible]),$TMP0 ;SAVE THE CURRENT "LPR"
4633  015676  004567  005214                      JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4634  015702  025144                              LINE
4635  015704  027607                              EM32+35
4636  015706  104032                              ERROR   32              ;AUTO ECHO TIMEOUT
4637  015710  000700                              BR      1$              ;GO TRY NEXT LINE
4638
4639  015712  010304                  4$:         MOV     R3,R4           ;EXTRACT LINE NUMBER OF RCVD CHAR
4640  015714  000304                              SWAB    R4
4641  015716  042704  177760                      BIC     #177760,R4
4642  015722  010402                              MOV     R4,R2           ;SAVE IT IN R2
4643  015724  006302                              ASL     R2              ;GENERATE TABLE INDEX IN R2
4644  015726  126704  007212                      CMPB    LINE,R4         ;IS THIS THE A.E. TEST LINE ??
4645  015732  001426                              BEQ     5$              ;BR IF YES
4646
4647  015734  026203  032436                      CMP     RBUF(R2),R3     ;RECVD DATA CORRECT ??
4648  015740  001447                              BEQ     6$              ;BR IF IT WAS
4649
4650  015742  004767  006366                      JSR     PC,SAPS         ;SAVE THE ERROR PSW
4651  015746  010467  163240                      MOV     R4,$TMP5        ;SAVE THE LINE NUMBER
4652  015752  016204  032436                      MOV     RBUF(R2),R4     ;SET UP S/B DATA
4653  015756  062702  032436                      ADD     #RBUF,R2        ;SET UP S/B ADDRESS
4654  015762  012701  177703                      MOV     #177703,R1      ;SET UP THE WAS ADDRESS
4655  015766  004767  004704                      JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4656  015772  004567  005120                      JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4657  015776  001212                              $TMP5
4658  016000  027452                              EM31+45
4659  016002  104031                              ERROR   31              ;NON-ECHO DATA COMPARE ERROR
4660  016004  000167  177502                      JMP     1$              ;GO TRY NEXT LINE
4661
4662
4663  016010  020367  163202          5$:         CMP     R3,$TMP7        ;CHAR ECHOED OK ??
4664  016014  001427                              BEQ     7$              ;BR IF YES
4665
4666  016016  004767  006312                      JSR     PC,SAPS         ;SAVE THE ERROR PSW
4667  016022  012702  001216                      MOV     #$TMP7,R2       ;SAVE THE S/B ADDRESS
4668  016026  016704  163164                      MOV     $TMP7,R4        ;SAVE THE S/B DATA
```

```
 4669  016032  012701  177703                    MOV     #177703,R1      ;SAVE THE WAS ADDRESS
 4670  016036  004767  004634                    JSR     PC,SL*R2A       ;GO SET UP ERROR INFO
 4671  016042  004567  005050                    JSR     R5,SUNUM        ;GO SET UP LINE NO. IN MESSAGE
 4672  016046  025144                            LINE
 4673  016050  027452                            EM31+45
 4674  016052  104031                            ERROR   31              ;AUTO ECHO LINE DATA ERROR
 4675  016054  000167  177432                    JMP     1$              ;GO TRY NEXT LINE
 4676
 4677  016060  105262  032436          6$:     INCB    RBUF(R2)        ;GENERATE NEXT EXPECTED DATA ON THIS LINE
 4678  016064  001266                            BNE     21$             ;BR IF ITS NOT BACK TO 000
 4679  016066  046267  024616  006562            BIC     LINBIT(R2),LINACT       ;INDICATE THIOS LINE DONE 256 BYTES
 4680  016074  005767  006556          7$:     TST     LINACT          ;ALL LINES INACTIVE
 4681  016100  001260                            BNE     21$             ;BR IF NOT
 4682  016102  042761  100000  000004            BIC     #BIT15,LPR(R1)  ;TURN OFF THE A.E. BIT
 4683  016110  105761  000017                    TSTB    SSR+1(R1)       ;SILO EMPTY ??
 4684  016114  001002                            BNE     8$              ;BR IF NOT
 4685  016116  000167  177370                    JMP     1$              ;GO TEST NEXT LINE
 4686  016122  000167  177514          8$:     JMP     21$             ;GO EMPTY IT
```

```
 4689                                   ;*********************************************************************
  (3)                                   ;*TEST 51       AUTO ECHO TEST 3 - ALL LINES
  (3)                                   ;*********************************************************************
  (2)  016126  000004                   TST51:  SCOPE
  (1)  016130  012767  000051  163076           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4690                                   .REM    %
  (1)                                   TEST ABSTRACT:
  (1)                                   **************
  (1)
  (1)                                           THIS TEST IS IDENTICAL TO TEST 47 EXCEPT ALL SELECTED LINES
  (1)                                   ARE ACTIVATED CONCURRENTLY RATHER THAN ONE AT A TIME AND ONLY
  (1)                                   THE ALL 1'S DATA IS USED.
  (1)
  (1)                                   ERRORS:
  (1)                                   *******
  (1)
  (1)                                   1. [ERROR  36] IS CALLED TO REPORT "DATA AVAIL" TIMEOUTS
  (1)                                   2. [ERROR  31] IS CALLED TO REPORT A.E. DATA ERRORS
  (1)
  (1)                                   SYNC:   M7277   SM4 LOAD BAR LB+HB L    CN2
  (1)                                   *****
  (1)
  (1)                                   DEBUG:
  (1)                                   ******
  (1)
  (1)                                           REFER TO TEST 47
  (1)
  (1)                                   KEY LOGIC:
  (1)                                   **********
  (1)
  (1)                                           REFER TO TEST 47
  (1)
  (1)                                   %
 4691  016136  012767  016144  162744           MOV     #1$,$LPERR      ;SET UP THE ERROR LOOP RETURN
 4692  016144  012711  004000          1$:      MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
 4693  016150  012705  000020                   MOV     #20,R5          ;INIT COUNTER TO SET UP 16. LINES
 4694  016154  012702  024516                   MOV     #AETAB,R2       ;SET UP POINTER TO AUTO ECHO TEST DATA
 4695  016160  012703  032376                   MOV     #RCNT,R3        ;R3 POINTS TO TABLE OF CHAR COUNTERS
 4696  016164  010261  000006          2$:      MOV     R2,CAR(R1)      ;SET UP BUS ADDRESS REG
 4697  016170  012761  177777  000010           MOV     #-1,BCR(R1)     ;SET UP BYTE COUNT REG
 4698  016176  012761  131403  000004           MOV     #131403,LPR(R1) ;SET UP LINE PARAMETERS
 4699  016204  005023                           CLR     (R3)+           ;CLEAR A COUNTER
 4700  016206  062702  000002                   ADD     #2,R2           ;UPDATE POINTERS
 4701  016212  005211                           INC     (R1)            ;SELECT NEXT LINE
 4702  016214  005305                           DEC     R5              ;COUNT ONE DONE
 4703  016216  001362                           BNE     2$              ;[ ] TILL 16. DONE
 4704  016220  116767  162656  162756           MOVB    $TSTNM,$TMP2    ;SAVE THE TEST NO.
 4705  016226  042767  177400  162750           BIC     #177400,$TMP2
 4706  016234  016767  006150  006414           MOV     LINSEL,LINACT   ;SET FLAG TO INDICATE ALL 16. ACTIVE
 4707  016242  016761  006142  000012           MOV     LINSEL,BAR(R1)  ;ACTIVATE ALL XMITTERS
 4708
 4709  016250  012767  000002  006714           MOV     #2,TIMEA        ;INIT TIMER A
 4710  016256  005067  006712                   CLR     TIMEB           ;INIT TIMERB
 4711  016262  105711                  3$:      TSTB    (R1)            ;CHAR AVAIL SET YET ?
```

```
4712  016264  100410                        BMI     4$              ;BR IF YES
4713  016266  004767  005660                JSR     PC,TIMEIT       ;CALL THE TIMER
4714  016272  000773                        BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4715                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4716
4717  016274  016167  000004  162676        MOV     LPR(R1),$TMP0   ;SAVE THE "LPR" REG
4718  016302  104036                        ERROR   36              ;DATA AVAILABLE TIMEOUT
4719  016304  000453                        BR      TST52           ;;EXIT TEST ON ERROR
4720
4721  016306  016103  000002          4$:   MOV     NRC(R1),R3      ;GET THE WAS DATA
4722  016312  010302                        MOV     R3,R2           ;BUILD AND SAVE LINE NO.
4723  016314  000302                        SWAB    R2
4724  016316  042702  177760                BIC     #177760,R2
4725  016322  010267  162666                MOV     R2,$TMP6        ;SAVE THE LINE NO.
4726  016326  006302                        ASL     R2              ;GENERATE TABLE OFFSET
4727  016330  005262  032376                INC     RCNT(R2)        ;COUNT THE CHARACTER
4728  016334  020362  024516                CMP     R3,AETAB(R2)    ;IS THE DATA CORRECT ??
4729  016340  001420                        BEQ     5$              ;BR IF YES
4730
4731  016342  004767  005766                JSR     PC,SAPS         ;SAVE THE ERROR PSW
4732  016346  016204  024516                MOV     AETAB(R2),R4    ;GET THE S/B DATA
4733  016352  062702  024516                ADD     #AETAB,R2       ;GENERATE S/B ADDRESS
4734  016356  062701  000002                ADD     #NRC,R1         ;GENERATE THE WAS ADDRESS
4735  016362  004767  004310                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4736  016366  004567  004524                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
4737  016372  001214                        $TMP6
4738  016374  027452                        EM31+45
4739  016376  104031                        ERROR   31              ;DATA COMPARE ERROR
4740  016400  000415                        BR      TST52           ;;EXIT TEST ON ERROR
4741
4742  016402  022762  000100  032376  5$:   CMP     #100,RCNT(R2)   ;DONE 64. CHARS ON THIS LINE ?
4743  016410  001324                        BNE     3$              ;BR IF NOT
4744  016412  016711  162576                MOV     $TMP6,(R1)      ;SELECT LINE IN SCR REG
4745  016416  042761  100000  000004        BIC     #BIT15,LPR(R1)  ;TURN OFF A.E. BIT
4746  016424  046267  024616  006224        BIC     LINBIT(R2),LINACT       ;ALL LINES INACTIVE ??
4747  016432  001313                        BNE     3$              ;BR IF NOT
```

```
                                ;*****************************************************************
                                ;*TEST 52       BREAK BIT TEST - ALL LINES
                                ;*****************************************************************
016434  000004                  TST52:  SCOPE
016436  012767  000052  162570          MOV     #STN-1,STESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                .REM    %
                                TEST ABSTRACT:
                                **************

                                        THIS TEST VERIFIES THAT THE "BREAK" FEATURE WORKS PROPERLY
                                FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS FOLLOWS:

                                        1. SET UP THE ERROR LOOP RETURN
                                        2. RETRIEVE THE CORRECT S/B DATA FROM THE "BREAK" DATA
                                           TABLE AND UPDATE THE POINTER.
                                        3. GO SELECT A LINE TO TEST - GO TO TEST 52 IF DONE ALL
                                           SELECTED LINES
                                        4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES.
                                        5. PRIME SELECTED LINE TO OUTPUT TWO "NULL" CHARS TO CLEAR UART
                                        6. ACTIVATE THE SELECTED LINE
                                        7. WAIT FOR SILO TO RECEIVE TWO NULLS - IF TIMEOUT REPORT ERROR
                                           AND RESTART AT STEP 2
                                        8. IF NO TIMEOUT CLEAR THE SELECTED DH11 AND RESELECT LINE NO.
                                        9. PRIME SELECTED LINE TO OUT PUT 256. CHARS.
                                        10.SET THE SELECTED LINE'S BREAK BIT
                                        11.ACTIVATE THE SELECTED LINE
                                        12.WAIT FOR "BAR" REG TO CLEAR -F TIMEOUT REPORT ERROR AND
                                           RESTART AT STEP 2
                                        13.IF NO TIMEOUT VERIFY THAT THE SILO RECEIVED ONLY ONE CHAR-
                                           IF NOT REPORT ERROR AND RESTART AT STEP 2
                                        14.IF SILO RECEIVED ONLY ONE CHAR VERIFY THAT IT WAS A "BREAK"
                                           CHAR - IF NOT REPORT ERROR - AND RESTART AT STEP 2

                                ERRORS:
                                ********

                                1. [ERROR  25] IS CALLED TO REPORT ALL ERRORS

                                SYNC:   M7277   SH4     LOAD BCR H      FU1
                                *****

                                DEBUG:
                                *******

                                1. IF ALL LINES FAILED SUSPECT THAT THE M7277 IS NOT GENERATING THE
                                   BREAK CONTROL REG LOAD SIGNAL.

                                2. IF ONLY ONE LINE FAILS SUSPECT THE BREAK CONTROL LOGIC ON THE M7278

                                KEY SIGNALS:
                                ************

                                        M7277   SH4     LOAD BCR H      FU1
```

```
(1)                                          M7278   SH5 THRU SH8
(1)
(1)                                                  74175 REGISTER CHIPS E51, E38, E67, E60
(1)                                                  7400 DRIVERS    E45, E46, E75, E76
(1)
(1)
(1)                                  X
4752  016444  012767  016524  162436          MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
4753  016452  012705  024720                  MOV     #BRKTAB,R5      ;SET UP POINTER TO BREAK DATA TABLE
4754  016456  005067  005732                  CLR     LMSK1           ;INIT BIT TEST MASK
4755  016462  000261                          SEC                     ;SET BIT MARKER IN "C"
4756  016464  000401                          BR      12$             ;GO SHIFT MASK
4757  016466  000241                  1$:     CLC                     ;INIT THE "C" BIT
4758  016470  006167  005720          12$:    ROL     LMSK1           ;SHIFT TEST MARKER
4759  016474  001411                          BEQ     11$             ;BR IF ALL LINES DONE
4760  016476  012504                          MOV     (R5)+,R4        ;GET TEST DATA FOR THIS LINE
4761  016500  036767  005710  005702          BIT     LMSK1,LINSEL    ;LINE SELECTED ?
4762  016506  001767                          BEQ     1$              ;BR IF NOT
4763  016510  004767  004310                  JSR     PC,SELINE       ;GO SELECT A LINE TO TEST
4764  016514  000401                          BR      11$             ;BR IF DONE ALL SELECTED LINES
4765  016516  000402                          BR      2$              ;GO TEST THE SELECTED LINE
4766  016520  000167  000454          11$:    JMP     9$              ;GO EXIT TEST
4767  016524  012711  004000          2$:     MOV     #BIT11,(R1)     ;CLEAR THE DH11
4768  016530  004767  004442                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
4769  016534  116711  006404                  MOVB    LINE,(R1)       ;SELECT THE LINE
4770
4771  016540  012761  025200  000006          MOV     #TNULL,CAR(R1)  ;SET UP TO OUTPUT TWO NULL CHARS
4772  016546  012761  177776  000010          MOV     #-2,BCR(R1)     ;SET BYTE COUNT TO 2
4773  016554  012761  033503  000004          MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
4774  016562  016761  005624  000012          MOV     LINMSK,BAR(R1)  ;ACTIVATE SELECTED LINE
4775
4776  016570  012767  000001  006374          MOV     #1,TIMEA        ;INIT TIMER A
4777  016576  005067  006372                  CLR     TIMEB           ;INIT TIMER B
4778  016602  122761  000002  000017  3$:     CMPB    #2,SSR+1(R1)    ;TWO CHARS RECEIVED ??
4779  016610  001432                          BEQ     4$              ;BR IF YES
4780  016612  004767  005334                  JSR     PC,TIMEIT       ;CALL THE TIMER
4781  016616  000771                          BR      3$              ;TIMER ROUTINE WILL MOVE RETURN PC
4782                                                                  ;AROUND THIS BRANCH IF TIME OUT OCCURS
4783
4784  016620  004767  005510                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
4785  016624  010467  162352                  MOV     R4,$TMP1        ;SAVE S/B DATA
4786  016630  010102                          MOV     R1,R2           ;SET UP REGADR
4787  016632  062702  000016                  ADD     #SSR,R2
4788  016636  011203                          MOV     (R2),R3         ;GET THE WAS DATA
4789  016640  042703  100377                  BIC     #100377,R3      ;CLEAR JUNK
4790  016644  012704  000002                  MOV     #2,R4           ;SET UP S/B DATA
4791  016650  000304                          SWAB    R4
4792  016652  004767  004020                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4793  016656  004567  004234                  JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
4794  016662  025144                          LINE
4795  016664  027155                          EM2S+34
4796  016666  016704  162310                  MOV     $TMP1,R4        ;RESTORE S/B DATA
4797  016672  104025                          ERROR   25              ;TIMED OUT WAITING FOR TWO NULLS
4798  016674  000674                          BR      1$              ;GO TRY NEXT LINE
4799
```

```
 4800  016676  012711  004000          4$:   MOV     #BIT11,(R1)     ;CLEAR THE INTERFACE
 4801  016702  116711  006236                MOVB    LINE,(R1)       ;SELECT THE LINE
 4802  016706  012761  033436  000006        MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDRESS REG FOR XMITTR
 4803  016714  012761  177400  000010        MOV     #-400,BCR(R1)   ;SET BYTE COUNT TO XMIT 256(10) CHARS
 4804  016722  012761  033503  000004        MOV     #33503,LPR(R1)  ;SET UP LINE PARAMETERS
 4805  016730  016761  005456  000014        MOV     LINMSK,BKR(R1)  ;SET BREAK BIT FOR ACTIVE LINE
 4806  016736  016761  005450  000012        MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4807
 4808  016744  012767  000005  006220        MOV     #5,TIMEA        ;INIT TIMER A
 4809  016752  005067  006216                CLR     TIMEB           ;INIT TIMER B
 4810  016756  005761  000012          5$:   TST     BAR(R1)         ;BAR BIT CLEARED ??
 4811  016762  001426                BEQ     6$              ;BR IFD YES
 4812  016764  004767  005162          JSR     PC,TIMEIT       ;CALL THE TIMER
 4813  016770  000772                BR      5$              ;TIMER ROUTINE WILL MOVE RETURN PC
 4814                                                        ;AROUND THIS BRANCH IF TIME OUT OCCURS
 4815
 4816  016772  004767  005336          JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4817  016776  010467  162200          MOV     R4,$TMP1        ;SAVE THE S/B DATA
 4818  017002  010102                MOV     R1,R2           ;SET UP REGADR
 4819  017004  062702  000012          ADD     #BAR,R2
 4820  017010  011203                MOV     (R2),R3         ;GET THE WAS DATA
 4821  017012  005004                CLR     R4              ;SET UP S/B DATA
 4822  017014  004767  003656          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4823  017020  004567  004072          JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
 4824  017024  025144                LINE
 4825  017026  027155                EM25+34
 4826  017030  016704  162146          MOV     $TMP1,R4        ;RESTORE THE S/B DATA
 4827  017034  104025                ERROR   25              ;BAR BIT FAILED TO CLEAR
 4828  017036  000613                BR      1$              ;GO TRY NEXT LINE
 4829
 4830  017040  122761  000001  000017  6$:   CMPB    #1,SSR+1(R1)    ;ONE CHAR RECEIVED ?
 4831  017046  001430                BEQ     7$              ;BR IF YES
 4832
 4833  017050  004767  005260          JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4834  017054  010467  162122          MOV     R4,$TMP1        ;SAVE THE S/B DATA
 4835  017060  010102                MOV     R1,R2           ;SET UP REGADR
 4836  017062  062702  000016          ADD     #SSR,R2
 4837  017066  011203                MOV     (R2),R3         ;GET THE WAS DATA
 4838  017070  042703  100377          BIC     #100377,R3      ;CLEAR JUNK
 4839  017074  012704  000001          MOV     #1,R4           ;SET UP S/B DATA
 4840  017100  000304                SWAB    R4
 4841  017102  004767  003570          JSR     PC,SUER2A       ;GO SET UP ERROR INFO
 4842  017106  004567  004004          JSR     R5,SUNUM        ;GO PUT LINE NO. IN MESSAGE
 4843  017112  025144                LINE
 4844  017114  027155                EM25+34
 4845  017116  016704  162060          MOV     $TMP1,R4        ;RESTORE THE S/B DATA
 4846  017122  104025                ERROR   25              ;FAILED TO RECEIVE THE ONE CHAR
 4847  017124  000167  177336          JMP     1$              ;GO TRY NEXT LINE
 4848
 4849
 4850  017130  016103  000002          7$:   MOV     NRC(R1),R3      ;GET THE WAS DATA
 4851  017134  020304                CMP     R3,R4           ;WAS IT A BREAK CHAR ?
 4852  017136  001002                BNE     8$              ;BR IF NOT CORRECT
 4853  017140  000167  177322          JMP     1$              ;GO TEST NEXT LINE
```

D03


```
4854
4855  017144  004767  005164          8$:     JSR     PC,SAPS         ;SAVE THE ERROR PSW
4856  017150  010102                          MOV     R1,R2           ;SET UP REGADR
4857  017152  062702  000002                  ADD     #NRC,R2
4858  017156  004767  003514                  JSR     PC,SUER2A       ;GO SET UP ERROR INFO
4859  017162  004567  003730                  JSR     R5,SUNUM        ;PUT LINE NO IN MESSAGE
4860  017166  025144                          LINE
4861  017170  027155                          EM25+34
4862  017172  104025                          ERROR   25              ;INCORRECT DATA RECEIVED
4863  017174  000167  177266                  JMP     1$              ;GO TRY NEXT LINE
4864  017200  000240                  9$:     NOP                     ;EXIT THIS TEST
4865
```

```
                                        ;*****************************************************************
                                        ;*TEST 53       HALF DUPLEX TEST - ALL LINES
                                        ;*****************************************************************
                                        TST53:  SCOPE
017202  000004
017204  012767  000053  162022                  MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
                                        .REM    X
                                        TEST ABSTRACT:
                                        *************

                                                THIS TEST VERIFIES THAT THE RECEIVERS ON ALL SELECTED LINES ARE
                                        "BLINDED" WHEN THE HALF-DUPLEX MODE IS ENABLED. THE TEST
                                        SEQUENCE IS AS FOLLOWS:

                                                1. SET UP THE ERROR LOOP RETURN
                                                2. GO SELECT A LINE NO. TO TEST
                                                3. IF DONE ALL SELECTED LINES - GO TO TEST 53
                                                4. RESET THE DHII AND CLEAR THE "CAR" AND "BCR" MEMORIES
                                                5. PRIME THE SELECTED DHII TO XMIT 256. CHARS IN HALF-
                                                   DUPLEX MODE.
                                                6. ACTIVATE THE SELECTED LINE AND WAIT FOR THE "BAR" REG TO CLEAR
                                                7. IF TIMEOUT - REPORT ERROR AND GO TO STEP 2
                                                8. IF NO TIMEOUT VERIFY THE "CHAR AVAIL" DID NOT SET (RECEIVER
                                                   BLINDED) - IF ERROR REPORT IT AND GO TO STEP2 - IF NO ERROR
                                                    GO TO STEP 2

                                        ERRORS:
                                        *******

                                        1. [ERROR  26] IS CALLED TO REPORT ALL ERRORS

                                        SYNC:   M7277   SH3     INIT A H        EF2
                                        *****

                                        DEBUG:
                                        ******

                                        1. SUSPECT EITHER THE M7289 OR THE M7288 MODULES

                                        KEY LOGIC:
                                        ***********

                                                M7289   SH5     HALF DUPLEX <15:00> H SIGNALS
                                                                END OF CHAR <15:00> SIGNALS

                                                M7288   SH5     HALF DUPLEX <03:00> H SIGNALS
                                                        SH7     "       "   <07:00> H SIGNALS
                                                        SH9     "       "   <11:08> H SIGNALS
                                                        SH11    "       "   <15:12> H SIGNALS

                                        %
017212  012767  017226  161670                  MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
017220  004767  003600          1$:     JSR     PC,$ELINE
017224  000477                          BR      TST54                   ;;BR IF ALL LINES TESTED
017226  012711  004000          2$:     MOV     #BIT11,(R1)             ;CLEAR THE INTERFACE
```

```
 4874  017232  004767  003240                  JSR     PC,CLCABC       ;GO CLR THE "CAR" AND "BCR" MEMORIES
 4875  017236  156711  000002                  BISB    LINE,(R1)       ;SELECT THE LINE
 4876  017242  016761  000006  000006          MOV     #TBUF,CAR(R1)   ;POINT TO XMIT BUFFER
 4877  017250  016761  177400  000010          MOV     #-400,BCR(R1)   ;XMIT 256(10) CHARS
 4878  017256  016761  073003  000004          MOV     #73003,LPR(R1)  ;SET UP THE LINE PARAMETERS
 4879  017264  016761  005122  000012          MOV     LINBSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE
 4880
 4881  017272  012767  000001  005672          MOV     #1,TIMEA        ;INIT TIME A
 4882  017300  005067  005670                  CLR     TIMEB           ;INIT TIME B
 4883  017304  005761  000012          3$:     TST     BAR(R1)         ;WAIT FOR XMITTR TO FINISH
 4884  017310  001423                          BEQ     4$              ;BR IF XMITTR FINISHED
 4885  017312  004767  004634                  JSR     PC,TIMEIT       ;CALL TIMER
 4886  017316  000772                          BR      3$              ;TIMER WILL MOVE RETURN PC AROUND
 4887                                                                  ;THIS BRANCH IF TIMEOUT OCCURS
 4888
 4889  017320  004767  005010                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4890  017324  016103  000012                  MOV     BAR(R1),R3      ;GET THE WAS DATA
 4891  017330  010102                          MOV     R1,R2           ;SET UP REGADR
 4892  017332  062702  000012                  ADD     #BAR,R2
 4893  017336  005004                          CLR     R4              ;SET UP NEW S/B DATA
 4894  017340  004767  003232                  JSR     PC,SUER2A       ;GO SET UP THE ERROR INFO
 4895  017344  004567  003216                  JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE
 4896  017350  025144                          LINE
 4897  017352  027220                          EM5+37
 4898  017354  104026                          ERROR   26              ;BAR BIT FAILED TO CLEAR ON TIME
 4899  017356  000720                          BR      1$              ;GO TRY NEXT LINE
 4900
 4901  017360  105711                  4$:     TSTB    (R1)            ;CHAR AVAIL SET ??
 4902  017362  100316                          BPL     1$              ;BR IF NOT IT SHOULDN'T BE
 4903
 4904  017364  004767  004744                  JSR     PC,SAPS         ;SAVE THE ERROR PSW
 4905  017370  010102                          MOV     R1,R2           ;SET UP REGADR
 4906  017372  011103                          MOV     (R1),R3         ;GET WAS DATA
 4907  017374  042703  100000                  BIC     #BIT15,R3       ;CLEAR JUNK BIT
 4908  017400  116704  005540                  MOVB    LINE,R4         ;SET UP S/B DATA
 4909  017404  004767  003266                  JSR     PC,SUER2A       ;GO SETUP ERROR INFO
 4910  017410  004567  003502                  JSR     R5,SUNUM        ;PUT LINE NO. IN MSG
 4911  017414  025144                          LINE
 4912  017416  027220                          EM26+37
 4913  017420  104026                          ERROR   26              ;HALF DUPLEX FAILED TO BLIND RECVR
 4914  017422  000676                          BR      1$              ;GO SELECT NEXT LINE
```

```
 4917                                   ;*****************************************************************
  (3)                                   ;*TEST 54       VERIFY THAT OVERRUN CAN SET PROPERLY - ALL LINES
  (3)                                   ;*****************************************************************
  (2)  017424  000004                   TST54:  SCOPE
  (1)  017426  012767  000054  161600           MOV     #STN-1,$TESTN   ;;SET TEST NUMBER IN  MAIL BOX
 4918                                   .REM    X
  (1)                                   TEST ABSTRACT:
  (1)                                   **************
  (1)
  (1)                                           THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
  (1)                                   THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
  (1)                                   PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:
  (1)
  (1)                                           1. SET UP THE ERROR LOOP RETURN
  (1)                                           2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
  (1)                                              END OF PASS HANDLER.
  (1)                                           3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
  (1)                                           4. ACTIVATE THE SELECTED LINE
  (1)                                           5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
  (1)                                              AND RESTART AT STEP 2
  (1)                                           6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
  (1)                                              "OVERRUN" IS SET ON THE LAST WORD READ
  (1)                                           7. IF NOT REPORT ERROR AND RESTART AT STEP 2
  (1)
  (1)                                   ERRORS:
  (1)                                   ********
  (1)
  (1)                                   1. [ERROR  50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
  (1)                                   2. [ERROR  56] IS CALLEDT TO REPORT "OVERRUN" ERROR
  (1)
  (1)                                   SYNC:   M7277   SH3     INIT A H        EF2
  (1)                                   *****
  (1)
  (1)                                   DEBUG:
  (1)                                   ******
  (1)
  (1)                                   1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
  (1)                                      FOR THE APPROPRIATE LINE IN QUESTION.
  (1)
  (1)                                   2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE
  (1)
  (1)                                   KEY LOGIC:
  (1)                                   **********
  (1)
  (1)                                           M7279   SH1     MASTER OR H     E12-9
  (1)                                                   SH2     MEMORY CHIP (3341) E13-11
  (1)
  (1)                                           M7280   SH2     UC1 OR 2 MASTER OR      EN2
  (1)                                                   SH2-5   UART PIN 15 (BUF OR LINE NN)
  (1)
  (1)                                   %
  (1)
 4919  017434  012767  017450  161446           MOV     #2$,$LPERR      ;SET UP ERROR LOOP RETURN
 4920  017442  004767  003356           1$:     JSR     PC,$ELINE       ;GO SELECT A LINE # TO TEST
```

```
                                                BR      ENDA            ;;BR IF DONE ALL SELECTED LINES
                                        2$:     MOV     #BIT11,(R1)     ;CLEAR OUT THE DH11
                                                MOVB    LINE,(R1)       ;SELECT THE LINE TO TEST
                                                MOV     #TBUF,CAR(R1)   ;SET UP CURRENT ADDRESS
                                                MOV     #-60.,BCR(R1)   ;SET UP BYTE COUNT REG
                                                MOV     #3[illegible]03,LPR(R1) ;DO IT AT 9600 BAUD - 8 BITS
                                                MOV     LINMSK,BAR(R1)  ;ACTIVATE THE SELECTED LINE

                                                MOV     #1,TIMEA        ;INIT TIMERS A AND B
                                                CLR     TIMEB
                                        3$:     TS      (R1)            ;TRANSMITTER DONE ??
                                                BMI     4$              ;BR IF YES
                                                JSR     PC,TIMEIT       ;CALL TIMER
                                                BR      3$              ;BR IF NO TIMEOUT

                                                JSR     PC,SAPS         ;GO SAVE PSW
                                                MOV     (R1),R3         ;GET THE WAS DATA
                                                BIC     #77760,R3       ;CLEAR UNINTERESTING BITS
                                                MOVB    LINE,R4         ;SET UP S/B DATA
                                                BIS     #BIT15,R4
                                                MOV     R1,R2           ;SET UP REGADR
                                                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
                                                JSR     R5,SUNUM        ;PUT LINE NO. IN MESSAGE HEADER
                                                LINE
                                                E[illegible]0+53
                                                ERROR   50              ;REPORT XMIT DONE TIME OUT
                                                BR      1$              ;GO TRY NEXT LINE

                                        4$:     MOV     #65.,$TMP1      ;SET UP TO READ 65. WORDS FROM SILO
                                                MOVB    LINE,R4         ;SET UP S/B DATA
                                                SWAB    R4
                                                BISB    #65.,R4
                                                BIS     #BIT15+BIT14,R4 ;PUT IN OVERRUN AND VALID DATA BITS
                                        5$:     MOV     RBUF(R1),R3     ;GET WAS DATA FROM SILO
                                                DEC     $TMP1           ;COUNT ONE WORD READ
                                                BNE     5$              ;BR TIL 65. READ
                                                CMP     R3,R4           ;WAS DATA AND OVERRUN CORRECT ??
                                                BEQ     1$              ;BR IF YES TRY NEXT SELECTED LINE

                                                JSR     PC,SAPS         ;GO SAVE PSW
                                                MOV     R1,R2           ;SET UP REGADR
                                                ADD     #RBUF,R2
                                                JSR     PC,SUER2A       ;GO SET UP ERROR INFO
                                                JSR     R5,SUNUM        ;GO PUT LINE NO. IN MSG HDR
                                                LINE
                                                E[illegible]+42
                                                ERROR   56              ;OVERRUN OR DATA INCORRECT
                                                BR      1$              ;GO TEST NEXT SELECTED LINE
```

```
 4972  017674  000004                    ENDA:   SCOPE
 4973  017676  012767  000240  000054            MOV     #240,$EOP       ;NOP THE SCOPE AT THE BEGINNING OF EOP
 4974  017704  005267  0[illegible]32            INC     DHNUM           ;GENERATE NEW DH11 NUMBER
 4975  017710  062767  000[illegible]2  00[illegible]32            ADD     #2,AOPTR        ;UPDATE THE TABLE POINTERS
 4976  017716  062767  000[illegible]2  0[illegible]6             ADD     #2,VCPTR
 4977  017724  062767  000[illegible]2  00[illegible]22            ADD     #2,BRPTR
 4978  017732  006367  004446                    ASL     SELMSK          ;SHIFT MARKER TO TEST NEXT DH11
 4979  017736  001410                    BEQ     $EOP            ;BR IF TESTED ALL SELECTED DH11'S
 4980  017740  036767  004440  004440            BIT     SELMSK,DHSEL    ;IS THIS DH11 SELECTED ?
 4981  017746  001752                    BEQ     ENDA            ;BR IF NOT
 4982  017750  105067  161126                    CLRB    $TSTNM          ;INIT TEST NUMBER
 4983  017754  000167  162632                    JMP     @STRTA          ;GO TEST THIS DH11
 4984                                    ;******************************************************************
  (1)
  (1)                                   .SBTTL  END OF PASS ROUTINE
  (1)
  (1)                                   ;*INCREMENT THE PASS NUMBER ($PASS)
  (1)                                   ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
  (1)                                   ;*IF THERES A MONITOR GO TO IT
  (1)                                   ;*IF THERE ISN'T JUMP TO START2
  (1)
  (1)  017760                           $EOP:
  (1)  017760  000004                            SCOPE
  (1)  017762  005[illegible]67  161114                    CLR     $TSTNM          ;;ZERO THE TEST NUMBER
  (1)  017766  005067  161226                    CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (1)  017772  00[illegible]267  161240                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
  (1)  017776  042767  100000  161232                    BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
  (1)  020004  005327                    DEC     (PC)+           ;;LOOP?
  (1)  020006  0[illegible]01                    $EOPCT: .WORD   1
  (1)  020010  0[illegible]22                    BGT     $DOAGN          ;;YES
  (1)  0[illegible]12  012737                    MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (1)  0[illegible]14  0[illegible]01                    $ENDCT: .WORD   1
  (1)  0[illegible]16  0[illegible]06                    $EOPCT
  (1)  0[illegible]20  104400  020062                    TYPE    ,$ENDMG         ;;TYPE "END PASS #"
  (2)  0[illegible]24  016746  161206                    MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
  (2)  0[illegible]30  104404                    TYPOS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
  (1)  0[illegible]32  104400  020077                    TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
  (1)  0[illegible]36                    $GET42:
  (1)
  (1)  0[illegible]36  013700  000042                    MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (1)  0[illegible]42  0[illegible]405                    BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  (1)  0[illegible]44  0[illegible]005                    RESET                   ;;CLEAR THE WORLD
  (1)  0[illegible]46  0[illegible]710                    $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (1)  0[illegible]50  0[illegible]240                    NOP                     ;;SAVE ROOM
  (1)  0[illegible]52  0[illegible]40                     NOP                     ;;FOR
  (1)  0[illegible]54  000240                    NOP                      ;ACT11
  (1)  0[illegible]56                    $DOAGN:
  (1)  0[illegible]56  000137  002506                    JMP     @#START2                ;;RETURN
  (1)  0[illegible]62  005015  047105  020104    $ENDMG: .ASCIZ  <15><12>/END PASS #/
  (1)  020070  040520  051523  021440
  (1)  020076     000
  (1)  020077     377     377     000    $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
 4985                                    ;******************************************************************
  (1)
```

```
(1)                                     .SBTTL  SCOPE HANDLER ROUTINE
(1)
(1)                                     ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)                                     ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1)                                     ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1)                                     ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                     ;*SW14=1        LOOP ON TEST
(1)                                     ;*SW11=1        INHIBIT ITERATIONS
(1)                                     ;*SW09=1        LOOP ON ERROR
(1)                                     ;*SW08=1        LOOP ON TEST IN SWR<7:0>
(1)                                     ;*CALL
(1)                                     ;*      SCOPE           ;;SCOPE=IOT
(1)
(1)  020102                             $SCOPE:
(3)  020102  005067  005036             CLR     LINE            ;INIT THE LINE NO. TO ZERO
(3)  020106  016701  004266             MOV     DHADR,R1        ;SET UP DEVADR IN R1
(1)  020112  032777  040000  161016  1$: BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
(1)  020120  001114                     BNE     $OVER           ;;YES IF SW14=1
(1)                                     ;*****START OF CODE FOR THE XOR TESTER*****
(1)  020122  000416                     $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(1)                                                                     ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(1)  020124  013746  000004             MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)  020130  012737  020150  000004     MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(1)  020136  005737  177060             TST     @#177060        ;;TIME OUT ON XOR?
(1)  020142  012637  000004             MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  020146  000463                     BR      $SVLAD          ;;GO TO THE NEXT TEST
(1)  020150  022626                     5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(1)  020152  012637  000004             MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  020156  000423                     BR      7$              ;;LOOP ON THE PRESENT TEST
(1)  020160                             6$:;*****END OF CODE FOR THE XOR TESTER*****
(1)  020160  032777  000400  160750     BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
(1)  020166  001404                     BEQ     2$              ;;BR IF NO
(1)  020170  127767  160742  160704     CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
(1)  020176  001465                     BEQ     $OVER           ;;BR IF YES
(1)  020200  105767  160677         2$: TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(1)  020204  001421                     BEQ     3$              ;;BR IF NO
(1)  020206  126767  160703  160667     CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
(1)  020214  101015                     BHI     3$              ;;BR IF NO
(1)  020216  032777  001000  160712     BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
(1)  020224  001404                     BEQ     4$              ;;BR IF NO
(1)  020226  016767  160656  160652  7$: MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
(1)  020234  000446                     BR      $OVER
(1)  020236  105067  160641         4$: CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(1)  020242  005067  160752             CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1)  020246  000415                     BR      1$              ;;ESCAPE TO THE NEXT TEST
(1)  020250  032777  004000  160660  3$: BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
(1)  020256  001011                     BNE     1$              ;;BR IF YES
(1)  020260  005767  160752             TST     $PASS           ;;IF FIRST PASS OF PROGRAM
(1)  020264  001406                     BEQ     1$              ;;      INHIBIT ITERATIONS
(1)  020266  005267  160612             INC     $ICNT           ;;INCREMENT ITERATION COUNT
(1)  020272  026767  160722  160604     CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
(1)  020300  002024                     BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
(1)  020302  012767  000001  160574  1$: MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
(1)  020310  016767  000052  160702     MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
```

```
   (1)  020316  105267  160560            $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
   (1)  020322  116767  160554  160704            MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
   (1)  020330  011667  160552                    MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
   (1)  020334  011667  160550                    MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
   (1)  020340  005067  160656                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
   (1)  020344  112767  000001  160543            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
   (1)  020352  016777  160524  160560  $OVER:   MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
   (1)  020360  016716  160522                    MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
   (1)  020364  000002                            RTI                     ;;FIXES PS
   (1)  020366  000010            $MXCNT: 10                              ;;MAX. NUMBER OF ITERATIONS
  4986                            ;*****************************************************************
   (1)
   (1)                            .SBTTL  ERROR HANDLER ROUTINE
   (1)
   (1)                            ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
   (1)                            ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
   (1)                            ;*AND GO TO $ERRTYP ON ERROR
   (1)                            ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
   (1)                            ;*SW15=1        HALT ON ERROR
   (1)                            ;*SW13=1        INHIBIT ERROR TYPEOUTS
   (1)                            ;*SW09=1        LOOP ON ERROR
   (1)                            ;*CALL
   (1)                            ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
   (1)
   (1)  020370                    $ERROR:
   (1)  020370  105267  160507    7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
   (1)  020374  001775                    BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
   (1)  020376  016777  160500  160534        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
   (1)  020404  005267  160502            INC     $ERTTL          ;;INC THE ERROR COUNT
   (1)  020410  011667  160502            MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
   (1)  020414  162767  000002  160474    SUB     #2,$ERRPC
   (1)  020422  117767  160470  160464    MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
   (1)  020430  032777  020000  160500    BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
   (1)  020436  001004                    BNE     20$             ;;SKIP TYPEOUTS
   (1)  020440  004767  000072            JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
   (1)  020444  104400  001225            TYPE    ,$CRLF
   (1)  020450                    20$:
   (1)  020450  122767  000001  160572        CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
   (1)  020456  001007                    BNE     2$              ;;NO,SKIP APT ERROR REPORT
   (1)  020460  116767  160430  000004        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
   (1)  020466  004767  001146            JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
   (1)  020472     000                21$: .BYTE   0
   (1)  020473     000                    .BYTE   0
   (1)  020474  000777                22$: BR      22$             ;;APT ERROR LOOP
   (1)  020476  005777  160434        2$:  TST     @SWR            ;;HALT ON ERROR
   (1)  020502  100001                     BPL     3$              ;;SKIP IF CONTINUE
   (1)  020504  000000                     HALT                    ;;HALT ON ERROR!
   (1)  020506  032777  001000  160422  3$:  BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
   (1)  020514  001402                     BEQ     4$              ;;BR IF NO
   (1)  020516  016716  160366             MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
   (1)  020522  005767  160474        4$:  TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
   (1)  020526  001402                     BEQ     5$              ;;BR IF NONE
   (1)  020530  016716  160466             MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
   (1)  020534                        5$:
```

```
  (1)  020534  000002                        RTI                         ;;RETURN
4987
  (1)
  (1)                                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
  (1)
  (1)                                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
  (1)                                        ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
  (1)                                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
  (1)
  (1)  020536                                $ERRTYP:
  (1)  020536  104400  001225                        TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020542  010046                                MOV     R0,-(SP)            ;;SAVE R0
  (1)  020544  005000                                CLR     R0                  ;;PICKUP THE ITEM INDEX
  (1)  020546  153700  001114                        BISB    @#$ITEMB,R0
  (1)  020552  001004                                BNE     1$                  ;;IF ITEM NUMBER IS ZERO, JUST
  (1)                                                                            ;;TYPE THE PC OF THE ERROR
  (2)  020554  016746  160336                        MOV     $ERRPC,-(SP)        ;;SAVE $ERRPC FOR TYPEOUT
  (2)                                                                            ;;ERROR ADDRESS
  (2)  020560  104401                                TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  020562  000426                                BR      6$                  ;;GET OUT
  (1)  020564  005300                        1$:     DEC     R0                  ;;ADJUST THE INDEX SO THAT IT WILL
  (1)  020566  006300                                ASL     R0                  ;;      WORK FOR THE ERROR TABLE
  (1)  020570  006300                                ASL     R0
  (1)  020572  006300                                ASL     R0
  (1)  020574  062700  001354                        ADD     #$ERRTB,R0          ;;FORM TABLE POINTER
  (1)  020600  012067  000004                        MOV     (R0)+,2$            ;;PICKUP "ERROR MESSAGE" POINTER
  (1)  020604  001404                                BEQ     3$                  ;;SKIP TYPEOUT IF NO POINTER
  (1)  020606  104400                                TYPE                        ;;TYPE THE "ERROR MESSAGE"
  (1)  020610  000000                        2$:     .WORD   0                   ;;"ERROR MESSAGE" POINTER GOES HERE
  (1)  020612  104400  001225                        TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020616  012067  000004                3$:     MOV     (R0)+,4$            ;;PICKUP "DATA HEADER" POINTER
  (1)  020622  001404                                BEQ     5$                  ;;SKIP TYPEOUT IF 0
  (1)  020624  104400                                TYPE                        ;;TYPE THE "DATA HEADER"
  (1)  020626  000000                        4$:     .WORD   0                   ;;"DATA HEADER" POINTER GOES HERE
  (1)  020630  104400  001225                        TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020634  011000                        5$:     MOV     (R0),R0             ;;PICKUP "DATA TABLE" POINTER
  (1)  020636  001004                                BNE     7$                  ;;GO TYPE THE DATA
  (1)  020640  012600                        6$:     MOV     (SP)+,R0            ;;RESTORE R0
  (1)  020642  104400  001225                        TYPE    ,$CRLF              ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  020646  000207                                RTS     PC                  ;;RETURN
  (2)                                        7$:
  (2)  020650  013046                                MOV     @(R0)+,-(SP)        ;;SAVE @(R0)+ FOR TYPEOUT
  (2)  020652  104401                                TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  020654  005710                                TST     (R0)                ;;IS THERE ANOTHER NUMBER?
  (1)  020656  001770                                BEQ     6$                  ;;BR IF NO
  (1)  020660  104400  020666                        TYPE    ,8$                 ;;TYPE TWO(2) SPACES
  (1)  020664  000771                                BR      7$                  ;;LOOP
  (1)  020666  020040      000       8$:     .ASCIZ  /  /                        ;;TWO(2) SPACES
  (1)          020672                        .EVEN
4988
  (1)                                        ;*****************************************************************
  (1)
  (1)                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
  (1)
  (1)                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
```

```
(1)                                     ;*OCTAL (ASCII) NUMBER AND TYPE IT.
(1)                                     ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOS                   ;;CALL FOR TYPEOUT
(1)                                     ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1)                                     ;*      .BYTE   M               ;;M=1 OR 0
(1)                                     ;*                                      ;;1=TYPE LEADING ZEROS
(1)                                     ;*                                      ;;0=SUPPRESS LEADING ZEROS
(1)                                     ;*
(1)                                     ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)                                     ;*$TYPOS OR $TYPOC
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPON                   ;;CALL FOR TYPEOUT
(1)                                     ;*
(1)                                     ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)                                     ;*CALL:
(1)                                     ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
(1)                                     ;*      TYPOC                   ;;CALL FOR TYPEOUT
(1)
(1)  020672  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
(1)  020676  116667  000001  000211            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
(1)  020704  112667  000207                     MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
(1)  020710  062716  000002                     ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
(1)  020714  000406                             BR      $TYPON
(1)  020716  112767  000001  000171    $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
(1)  020724  112767  000006  000165            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
(1)  020732  112767  000005  000154    $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
(1)  020740  010346                             MOV     R3,-(SP)        ;;SAVE R3
(1)  020742  010446                             MOV     R4,-(SP)        ;;SAVE R4
(1)  020744  010546                             MOV     R5,-(SP)        ;;SAVE R5
(1)  020746  116704  000145                     MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
(1)  020752  005404                             NEG     R4
(1)  020754  062704  000006                     ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
(1)  020760  110467  000132                     MOVB    R4,$OMODE       ;;SAVE IT FOR USE
(1)  020764  116704  000125                     MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
(1)  020770  016605  000012                     MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
(1)  020774  005003                             CLR     R3              ;;CLEAR THE OUTPUT WORD
(1)  020776  006105                  1$:        ROL     R5              ;;ROTATE MSB INTO "C"
(1)  021000  000404                             BR      3$              ;;GO DO MSB
(1)  021002  006105                  2$:        ROL     R5              ;;FORM THIS DIGIT
(1)  021004  006105                             ROL     R5
(1)  021006  006105                             ROL     R5
(1)  021010  010503                             MOV     R5,R3
(1)  021012  006103                  3$:        ROL     R3              ;;GET LSB OF THIS DIGIT
(1)  021014  105367  000076                     DECB    $OMODE          ;;TYPE THIS DIGIT?
(1)  021020  100016                             BPL     7$              ;;BR IF NO
(1)  021022  042703  177770                     BIC     #177770,R3      ;;GET RID OF JUNK
(1)  021026  001002                             BNE     4$              ;;TEST FOR 0
(1)  021030  005704                             TST     R4              ;;SUPPRESS THIS 0?
(1)  021032  001403                             BEQ     5$              ;;BR IF YES
(1)  021034  005204                  4$:        INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  021036  052703  000060                     BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
```

```
(1)  021042  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  021046  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  021052  104400  021112                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  021056  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
(1)  021062  003347                          BGT     2$              ;;BR IF MORE TO DO
(1)  021064  002402                          BLT     6$              ;;BR IF DONE
(1)  021066  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  021070  000744                          BR      2$              ;;GO DO THE LAST DIGIT
(1)  021072  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)  021074  012604                          MOV     (SP)+,R4        ;;RESTORE R4
(1)  021076  012603                          MOV     (SP)+,R3        ;;RESTORE R3
(1)  021100  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  021106  012616                          MOV     (SP)+,(SP)
(1)  021110  000002                          RTI                     ;;RETURN
(1)  021112     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  021113     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  021114     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  021115     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)  021116  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
4989                                 ;*****************************************************************
(1)
(1)                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(1)                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1)                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(1)                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(1)                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(1)                                  ;*REPLACED WITH SPACES.
(1)                                  ;*CALL:
(1)                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
(1)                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
(1)
(2)  021120                          $TYPDS:
(3)  021120  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  021122  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  021124  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  021126  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  021130  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  021132  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
(1)  021136  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
(1)  021142  100004                          BPL     1$              ;;BR IF INPUT IS POS.
(1)  021144  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
(1)  021146  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
(1)  021154  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
(1)  021156  012703  021334                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
(1)  021162  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
(1)  021166  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
(1)  021170  016001  021324                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
(1)  021174  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
(1)  021176  002402                          BLT     4$              ;;BR IF DONE
(1)  021200  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
(1)  021202  000774                          BR      3$
(1)  021204  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
(1)  021206  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
```

```
(1)  021210  001002                        BNE     5$              ;;FALL THROUGH IF 0
(1)  021212  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
(1)  021214  100407                        BMI     7$              ;;BR IF YES
(1)  021216  106316                5$:     ASLB    (SP)            ;;MSD?
(1)  021220  103003                        BCC     6$              ;;BR IF NO
(1)  021222  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
(1)  021230  052702  000060        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
(1)  021234  052702  000040        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)  021240  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)  021242  005720                        TST     (R0)+           ;;JUST INCREMENTING
(1)  021244  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
(1)  021250  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
(1)  021252  003002                        BGT     8$              ;;GO TO EXIT
(1)  021254  010502                        MOV     R5,R2           ;;GET THE LSD
(1)  021256  000764                        BR      6$              ;;GO CHANGE TO ASCII
(1)  021260  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
(1)  021262  100003                        BPL     9$              ;;BR IF NO
(1)  021264  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
(1)  021272  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
(3)  021274  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  021276  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  021300  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  021302  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  021304  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  021306  104400  021334                TYPE    $DBLK           ;;NOW TYPE THE NUMBER
(1)  021312  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  021320  012616                        MOV     (SP)+,(SP)
(1)  021322  000002                        RTI                     ;;RETURN TO USER
(1)  021324  023420                $DTBL:  10000.
(1)  021326  001750                        1000.
(1)  021330  000144                        100.
(1)  021332  000012                        10.
(1)  021334  000004                $DBLK:  .BLKW   4
4990                                ;*****************************************************************
(1)
(1)                                 .SBTTL  TYPE ROUTINE
(1)
(1)                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                 ;*
(1)                                 ;*CALL:
(1)                                 ;*1) USING A TRAP INSTRUCTION
(1)                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                 ;*OR
(1)                                 ;*      TYPE
(1)                                 ;*      MESADR
(1)                                 ;*
(1)
(1)  021344  105767  157605        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  021350  100002                        BPL     1$              ;;BR IF YES
(1)  021352  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
```

```
(1)  021354  000430                        BR      3$              ;;LEAVE
(1)  021356  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  021360  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
(1)  021364  122767  000001  157656        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  021372  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
(1)  021374  132767  000100  157647        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
(1)  021402  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
(1)  021404  010067  000004                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
(1)  021410  004767  000214                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
(1)  021414  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
(1)  021416  132767  000040  157625  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
(1)  021424  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
(1)  021426  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  021430  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  021432  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  021434  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
(1)  021436  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  021442  000002                        RTI                     ;;RETURN
(1)  021444  122716  000011        4$:     CMPB    #THT,(SP)       ;;BRANCH IF <HT>
(1)  021450  001426                        BEQ     8$
(1)  021452  122716  000200                CMPB    #TCRLF,(SP)     ;;BRANCH IF NOT <CRLF>
(1)  021456  001004                        BNE     5$
(1)  021460  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
(1)  021462  104400                        TYPE                    ;;TYPE A CR AND LF
(1)  021464  001225                        $CRLF
(1)  021466  000757                        BR      2$              ;;GET NEXT CHARACTER
(1)  021470  004767  000056        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1)  021474  126726  157454        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1)  021500  001352                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1)  021502  016746  157444                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                                 ;;AND THE NULL CHAR.
(1)  021506  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1)  021512  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  021514  004767  000032                JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1)  021520  105367  000072                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  021524  000770                        BR      7$              ;;LOOP
(1)
(1)                                ;HORIZONTAL TAB PROCESSOR
(1)
(1)  021526  112716  000040        8$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
(1)  021532  004767  000014        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  021536  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  021544  001372                        BNE     9$              ;;TAB STOP
(1)  021546  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
(1)  021550  000726                        BR      2$              ;;GET NEXT CHARACTER
(1)  021552  105777  157370        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  021556  100375                        BPL     $TYPEC
(1)  021560  116677  000002  157362        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  021566  122766  000015  000002        CMPB    #15,2(SP)       ;;BRANCH IF
(1)  021574  001003                        BNE     1$              ;;NOT <CR>
(1)  021576  105067  000014                CLRB    $CHARCNT        ;;
(1)  021602  000406                        BR      $TYPEX          ;;EXIT
(1)  021604  122766  000012  000002  1$:    CMPB    #12,2(SP)       ;;BRANCH IF
(1)  021612  002002                        BGE     $TYPEX          ;;<LF>
```

```
(1)  021614  105227                        INCB    (PC)+           ;;INC SPACE
(1)  021616  000000                $CHARCNT:.WORD  0               ;;COUNT
(1)  021620  000207                $TYPEX: RTS     PC
(1)                                ;       EQUATES
(1)          000011                THT=11
(1)          000200                TCRLF=200
(1)
4991                               ;*****************************************************************

(1)                                .SBTTL  APT COMMUNICATIONS ROUTINE
(1)  021622  112767  000001  000236 $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
(1)  021630  112767  000001  000226 $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
(1)  021636  000403                        BR      $ATYC
(1)  021640  112767  000001  000220 $ATY4:  MOVB    #1,$FFLG        ;TO ONLY REPORT FATAL ERROR
(2)  021646                        $ATYC:
(3)  021646  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  021650  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(1)  021652  105767  000206                TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
(1)  021656  001450                        BEQ     5$              ;IF NOT:  BR
(1)  021660  122767  000001  157362        CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
(1)  021666  001031                        BNE     3$              ;IF NOT:  BR
(1)  021670  132767  000100  157353        BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGES?
(1)  021676  001425                        BEQ     3$              ;IF NOT:  BR
(1)  021700  017600  000004                MOV     @4(SP),R0       ;GET MESSAGE ADDR.
(1)  021704  062766  000002  000004        ADD     #2,4(SP)                ;BUMP RETURN ADDR.
(1)  021712  005767  157312          1$:   TST     $MSGTYPE        ;SEE IF DONE W/ LAST XMISSION?
(1)  021716  001375                        BNE     1$              ;IF NOT:  WAIT
(1)  021720  010067  157320                MOV     R0,$MSGAD       ;PUT ADDR IN MAILBOX
(1)  021724  105720                  2$:   TSTB    (R0)+           ;FIND END OF MESSAGE
(1)  021726  001376                        BNE     2$
(1)  021730  166700  157310                SUB     $MSGAD,R0       ;SUB START OF MESSAGE
(1)  021734  006200                        ASR     R0              ;GET MESSAGE LNGTH IN WORDS
(1)  021736  010067  157304                MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
(1)  021742  012767  000004  157260        MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MSG.
(1)  021750  000413                        BR      5$
(1)  021752  017667  000004  000016  3$:   MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
(1)  021760  062766  000002  000004        ADD     #2,4(SP)                ;BUMP RETURN ADDRESS
(3)  021766  016746  156004                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
(1)  021772  004767  177346                JSR     PC,$TYPE        ;CALL TYPE MACRO
(1)  021776  000000                  4$:   .WORD   0
(1)  022000                        5$:
(1)  022000  105767  000062          10$:  TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
(1)  022004  001416                        BEQ     12$             ;IF NOT:  BR
(1)  022006  005767  157236                TST     $ENV            ;RUNNING UNDER APT?
(1)  022012  001413                        BEQ     12$             ;IF NOT:  BR
(1)  022014  005767  157210          11$:  TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
(1)  022020  001375                        BNE     11$             ;IF NOT:  WAIT
(1)  022022  017667  000004  157202        MOV     @4(SP),$FATAL   ;GET ERROR #
(1)  022030  062766  000002  000004        ADD     #2,4(SP)                ;BUMP RETURN ADDR.
(1)  022036  005267  157166                INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
(1)  022042  105067  000020          12$:  CLRB    $FFLG           ;CLEAR FATAL FLAG
(1)  022046  105067  000013                CLRB    $LFLG           ;CLEAR LOG FLAG
(1)  022052  105067  000006                CLRB    $MFLG           ;CLEAR MESSAGE FLAG
(3)  022056  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
(3)  022060  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  022062  000207                        RTS     PC              ;RETURN
(1)
(1)  022064  000                   $MFLG:  .BYTE   0               ;MESSG. FLAG
(1)  022065  000                   $LFLG:  .BYTE   0               ;LOG FLAG
(1)  022066  000                   $FFLG:  .BYTE   0               ;FATAL FLAG
(1)          022070                        .EVEN
(1)          000200                APTSIZE=200
(1)          000001                APTENV=001
(1)          000100                APTSPOOL=100
(1)          000040                APTCSUP=040
4992                               ;*****************************************************************
(1)
(1)                                .SBTTL  TTY INPUT ROUTINE
(1)
(1)                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                ;*CALL:
(1)                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                ;
(1)
(1)  022070  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  022072  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  022100  105777  157036        1$:     TSTB    @$TKS           ;;WAIT FOR
(1)  022104  100375                        BPL     1$              ;;A CHARACTER
(1)  022106  117766  157032  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  022114  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  022122  000002                        RTI                     ;;GO BACK TO USER
(2)                                ;*****************************************************************
(1)                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                ;*CALL:
(1)                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1)  022124  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  022126  012703  022232        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  022132  022703  022242        2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
(1)  022136  101405                        BLOS    4$              ;;BR IF YES
(1)  022140  104405                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  022142  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  022144  122713  000177                CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  022150  001003                        BNE     3$              ;;SKIP IF NOT
(1)  022152  104400  001224        4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  022156  000763                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  022160  111367  000044        3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  022164  104400  022230                TYPE    ,9$
(1)  022170  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  022174  001356                        BNE     2$              ;;LOOP IF NOT RETURN
(1)  022176  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  022202  104400  001226                TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  022206  012603                        MOV     (SP)+,R3        ;;RESTORE R3
(1)  022210  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  022212  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
```

```
(1)  022220  012766  022232  000004          MOV     #STTYIN,4(SP)
(1)  022226  000002                          RTI                     ;;RETURN
(1)  022230     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  022231     000                          .BYTE   0               ;;TERMINATOR
(1)  022232  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
4993                                 ;*******************************************************************
(1)                                  
(1)                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
(1)                                  
(1)                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1)                                  ;*CHANGE IT TO BINARY.
(1)                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
(1)                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
(1)                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED, THE COMPLETE NUMBER MUST
(1)                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
(1)                                  ;*CALL:
(1)                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
(1)                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
(1)                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
(1)                                  
(1)  022242  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
(1)  022244  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
(3)  022252  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  022254  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  022256  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(1)  022260  104406                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
(1)  022262  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
(1)  022264  010067  000100                  MOV     R0,5$           ;;AND SAVE IT
(1)  022270  005001                          CLR     R1              ;;CLEAR DATA WORD
(1)  022272  005002                          CLR     R2
(1)  022274  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
(1)  022276  001420                          BEQ     3$              ;;IF ZERO GET OUT
(1)  022300  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
(1)  022304  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
(1)  022306  122716  000067                  CMPB    #'7,(SP)
(1)  022312  002423                          BLT     4$
(1)  022314  006301                          ASL     R1              ;;*2
(1)  022316  006102                          ROL     R2
(1)  022320  006301                          ASL     R1              ;;*4
(1)  022322  006102                          ROL     R2
(1)  022324  006301                          ASL     R1              ;;*8
(1)  022326  006102                          ROL     R2
(1)  022330  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
(1)  022334  060116                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
(1)  022336  000756                          BR      2$              ;;LOOP
(1)  022340  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
(1)  022342  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
(1)  022346  010267  000026                  MOV     R2,$HIOCT
(3)  022352  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  022354  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  022356  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  022360  000002                          RTI                     ;;RETURN
(1)  022362  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
(1)  022364  105010                          CLRB    (R0)            ;;SET A TERMINATOR
```

```
  (1)  022366  104400                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
  (1)  022370  000000              5$:       .WORD   0
  (1)  022372  104400  001224                TYPE    $QUES           ;;"?" "CR" & "LF"
  (1)  022376  000730                        BR      1$              ;;TRY AGAIN
  (1)  022400  000000              $HIOCT:   .WORD   0               ;;HIGH ORDER BITS GO HERE
 4994                                ;*****************************************************************
  (1)
  (1)                                .SBTTL  TRAP DECODER
  (1)
  (1)                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
  (1)                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
  (1)                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
  (1)                                ;*GO TO THAT ROUTINE.
  (1)
  (1)  022402  010046              $TRAP:    MOV     R0,-(SP)        ;;SAVE R0
  (1)  022404  016600  000002               MOV     2(SP),R0        ;;GET TRAP ADDRESS
  (1)  022410  005740                        TST     -(R0)           ;;BACKUP BY 2
  (1)  022412  111000                        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
  (1)  022414  006300                        ASL     R0              ;;POSITION FOR INDEXING
  (1)  022416  016000  022424                MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
  (1)  022422  000200                        RTS     R0              ;;GO TO ROUTINE
  (1)
  (3)
  (3)                                .SBTTL  TRAP TABLE
  (3)
  (3)                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
  (3)                                ;*BY THE "TRAP" INSTRUCTION.
  (3)
  (3)                                ;       ROUTINE
  (3)                                ;       -------
  (3)  022424              $TRPAD:
  (3)  022424  021344                $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
  (3)  022426  020716                $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
  (3)  022430  020672                $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
  (3)  022432  020732                $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
  (3)  022434  021120                $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
  (3)  022436  022070                $RDCHR  ;;CALL=RDCHR    TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
  (3)  022440  022124                $RDLIN  ;;CALL=RDLIN    TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
  (3)  022442  022242                $RDOCT  ;;CALL=RDOCT    TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
 4995                                ;*****************************************************************
  (1)
  (1)                                .SBTTL  POWER DOWN AND UP ROUTINES
  (1)
  (1)                                ;POWER DOWN ROUTINE
  (1)  022444  012737  022572  000024  $PWRDN:  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
  (1)  022452  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
  (3)  022460  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  022462  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  022464  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  022466  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  022470  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
  (3)  022472  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (1)  022474  010667  000076                MOV     SP,$SAVR6       ;;SAVE SP
  (1)  022500  012737  022512  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
```

```
(1)  022506  000000                          HALT
(1)  022510  000776                          BR      .-2             ;;HANG UP
(1)
(1)                                  ;POWER UP ROUTINE
(1)  022512  016706  000060          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
(1)  022516  005067  000054                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
(1)  022522  005267  000050          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
(1)  022526  001375                          BNE     1$              ;;OF  WORD
(3)  022530  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  022532  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
(3)  022534  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  022536  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  022540  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  022542  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  022544  012737  022444  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1)  022552  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
(1)  022560  104400                          TYPE                    ;REPORT THE POWER FAILURE
(1)  022562  022600                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
(1)  022564  012716                          MOV     (PC)+,(SP)      ;;RESTART AT RSTRTA
(1)  022566  002612                  $PWRAD: .WORD   RSTRTA          ;;RESTART ADDRESS
(1)  022570  000002                          RTI
(1)  022572  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
(1)  022574  000776                          BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
(1)  022576  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
(1)  022600  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
(1)  022606  000122
(1)                                          .EVEN
```

```
                                ;*****************************************************************
                                ;COMMON DH11 SERVICE ROUTINES
                                ;*****************************************************************

                                ;THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
                                ;OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN

022610  012701  033436        LDTBF1: MOV     #TBUF,R1        ;POINT TO START OF BUFFER
022614  005002                        CLR     R2              ;INIT DATA BYTE GENERATOR
022616  110221                1$:     MOVB    R2,(R1)+        ;LOAD ONE CHAR
022620  005202                        INC     R2              ;GENERATE NEXT CHAR
022622  022702  000400                CMP     #400,R2         ;LOADED 256(10) BYTES
022626  001373                        BNE     1$              ;BR IF NOT
022630  000207                        RTS     PC              ;RETURN TO START TESTING

                                ;THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
                                ;USING A "DH1" HEADER

022632  004767  001476        SUER1:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
022636  116700  156240                MOVB    $TSTNM,R0       ;SAVE THE TEST NO.
022642  010067  156312                MOV     R0,$REG0        ;SAVE THE TEST NO. FOR ERROR PRINT
022646  010167  156310                MOV     R1,$REG1        ;SAVE THE DH11 ADDR
022652  010267  156306                MOV     R2,$REG2        ;SAVE THE REG ADDRESS
022656  010667  156312                MOV     R6,$REG6        ;SAVE THE SP
022662  062767  000002  156304        ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
022670  000207                        RTS     PC              ;RETURN TO CALLING ROUTINE

                                ;THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
                                ;SAVE THE ERROR INFORMATION IN "DT2"

022672  004767  001436        SUER2:  JSR     PC,SAPS         ;SAVE THE ERROR PSW
022676  116700  156200        SUER2A: MOVB    $TSTNM,R0       ;GET THE TEST NO.
022702  010067  156252                MOV     R0,$REG0        ;SAVE THE REGISTERS-TEST#
022706  010167  156250                MOV     R1,$REG1        ;SAVE THE DH ADDRESS
022712  010267  156246                MOV     R2,$REG2        ;SAVE THE REGISTER ADDRESS
022716  010367  156244                MOV     R3,$REG3        ;SAVE THE WRS DATA
022722  010467  156242                MOV     R4,$REG4        ;SAVE THE S/B DATA
022726  010667  156242                MOV     R6,$REG6        ;SAVE THE STACK POINTER
022732  062767  000002  156234        ADD     #2,$REG6        ;CORRECT FOR CALLING JSR
022740  000207                        RTS     PC              ;RETURN TO REPORT ERROR

                                ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                ;BUS ERROR AND RSVD INSTR ERROR ROUTINES

022742  010067  156212        SUER3:  MOV     R0,$REG0        ;SAVE THE REGS
022746  010167  156210                MOV     R1,$REG1
022752  010267  156206                MOV     R2,$REG2
022756  000207                        RTS     PC              ;RETURN TO REPORT ERROR

                                ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
                                ;CAR/BCR MEMORY PATTERNS TESTS

022760  005067  156214        SUER4:  CLR     $TMP0           ;SAVE THE LINE NO. WRITTEN
022764  116767  002156  156206        MOVB    LINEA,$TMP0
```

```
5052  022772  116700  156104                        MOVB    $TSTNM,R0       ;SAVE THE TEST NUMBER
5053  022776  010067  156156                        MOV     R0,$REG0        ;SAVE THE REGISTER INFORMATION
5054  023002  010167  156154                        MOV     R1,$REG1
5055  023006  010267  156152                        MOV     R2,$REG2
5056  023012  010367  156150                        MOV     R3,$REG3
5057  023016  010467  156146                        MOV     R4,$REG4
5058  023022  000207                                RTS     PC              ;RETURN TO PATTERNS TEST
5059
5060                                  ;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
5061                                  ;VALUE OF THE LINE SELECTION PARAMETER
5062
5063                                          ;CALLING SEQUENCE:
5064
5065                                          ;JSR    PC,SELINE       ;CALL THE ROUTINE
5066                                          ;BR     1$              ;EXIT BRANCH-ROUTINE MOVES THE RETURN
5067                                                                  ;PC AROUND THIS BR IF MORE LINES ARE
5068                                                                  ;YET TO BE TESTED
5069
5070  023024  105767  002115        SELINE: TSTB    LINE+1          ;FIRST TIME THROUGH FOR ANY TEST ?
5071  023030  001010                        BNE     1$              ;BR IF NOT
5072  023032  105167  002107                COMB    LINE+1          ;SET ENTRY FLAG
5073  023036  012767  000001  001346        MOV     #1,LINMSK       ;INIT SELECT TEST MASK TO TEST LINE 00
5074  023044  105067  002074                CLRB    LINE            ;START WITH LINE #00
5075  023050  000405                        BR      2$              ;GO TEST FOR LINE #00
5076  023052  105267  002066        1$:     INCB    LINE            ;GENERATE NEW LINE NO.
5077  023056  006367  001330                ASL     LINMSK          ;SHIFT SELECT MASK TO TEST NXT LINE
5078  023062  001407                        BEQ     3$              ;RETURN TO EXIT BRANCH - ALL LINES DONE
5079  023064  036767  001322  001316  2$:     BIT     LINMSK,LINSEL   ;IS THE LINE SELECTED FOR TEST ??
5080  023072  001767                        BEQ     1$              ;BR IF NOT
5081  023074  062716  000002                ADD     #2,(SP)         ;MOVE RETURN PC AROUND EXIT BRANCH
5082  023100  000402                        BR      4$              ;RETURN TO TEST SELECTED LINE
5083  023102  005067  002036        3$:     CLR     LINE            ;INIT ENTRY FLAG AND LINE NO. TO 000
5084  023106  142777  000017  001264  4$:     BICB    #17,@DHRDR      ;INIT LINE SELECT BITS IN "SCR"
5085  023114  000207                        RTS     PC              ;RETURN TO CALLING TEST
5086
5087                                  ;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
5088                                  ;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
5089                                  ;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING
5090
5091                                          ;CALLING SEQUENCE
5092
5093                                          ;JSR    R5,SUNUM        ;CALL TO THIS ROUTINE
5094                                          ;ADDR1                  ;ADDRESS OF THE NUMBER TO BE CONVERTED
5095                                          ;ADDR2                  ;ADDRESS OF THE MSG BUFFER SLOT
5096
5097  023116                          SUNUM:
 (2)  023116  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (2)  023120  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (2)  023122  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
5098  023124  012500                        MOV     (R5)+,R0        ;GET ADDRESS OF NUMBER
5099  023126  012501                        MOV     (R5)+,R1        ;GET MSG BUFFER ADDR
5100  023130  111000                        MOVB    (R0),R0         ;GET NO. TO BE CONVERTED
5101  023132  010002                        MOV     R0,R2           ;SAVE IT IN R2
5102  023134  006202                        ASR     R2              ;SHIFT MSD TO LSD POSITION
```

```
5103  023134  006202                          ASR     R2
      023136  006202                          ASR     R2
      023140  042702  177770                  BIC     #177770,R2      ;CLR JUNK BITS
      023144  062702  000060                  ADD     #60,R2          ;MAKE IT ASCII
      023150  110221                          MOVB    R2,(R1)+        ;PUT IT IN MSG BUFFER
      023152  042700  177770                  BIC     #177770,R0      ;CLR JUNK FROM LSD
      023156  062700  000060                  ADD     #60,R0          ;MAKE IT ASCII
      023162  110011                          MOVB    R0,(R1)         ;PUT LSD IN THE BUFFER
5111  023164  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
  (2) 023166  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
  (2) 023170  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5112  023172  000205                          RTS     R5              ;RETURN TO CALLER
5113
5114                                  ;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
5115                                  ;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1
5116
5117  023176  005067  156014          CLCABC: CLR     $TMP7           ;INIT A COUNTER
5118  023202  116711  156010          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
5119  023206  005061  000006                  CLR     CAR(R1)         ;CLEAR A CAR LOCATION
5120  023212  005061  000010                  CLR     BCR(R1)         ;CLEAR A BCR LOCATION
5121  023216  005267  155774                  INC     $TMP7           ;GENERATE NEW LINE NO.
5122  023222  022767  000020  155766          CMP     #20,$TMP7       ;DONE ALL LINES ?
5123  023230  001364                          BNE     1$              ;BR IF NOT
5124  023232  142711  000017                  BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
5125  023236  000207                          RTS     PC              ;RETURN TO CALLER
5126
5127                                  ;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
5128                                  ;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1
5129
5130  023240  005067  155752          LDBCR:  CLR     $TMP7           ;INIT A COUNTER
5131  023244  116711  155746          1$:     MOVB    $TMP7,(R1)      ;SELECT A LINE
5132  023250  012761  177777  000010          MOV     #-1,BCR(R1)     ;LOAD BCR LOC. WITH 177777
5133  023256  005267  155734                  INC     $TMP7           ;GENERATE NEXT LINE NO.
5134  023262  022767  000020  155726          CMP     #20,$TMP7       ;DONE ALL LINES ?
5135  023270  001365                          BNE     1$              ;BR IF NOT
5136  023272  142711  000017                  BICB    #17,(R1)        ;SET "SCR" TO SELECT LINE 00
5137  023276  000207                          RTS     PC              ;RETURN TO CALLER
5138
5139                                  ;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS
5140
5141  023300  012767  000020  155702  SUPPAR: MOV     #20,$TMP4       ;SET UP FOR 16. LINES
5142  023306  105011                          CLRB    (R1)            ;INIT SCR TO START AT LINE 00
5143  023310  005002                          CLR     R2              ;INIT INDEX REGISTER FOR RBUF (EVEN)
5144  023312  012703  000200                  MOV     #200,R3         ;SET UP CONSTANT
5145  023316  012704  000001                  MOV     #1,R4           ;INIT INDEX REG FOR RBUF (ODD)
5146  023322  012761  033436  000006  1$:     MOV     #TBUF,CAR(R1)   ;LOAD BUS ADDRESS REWG
5147  023330  016761  155660  000010          MOV     $TMP6,BCR(R1)   ;LOAD BYTE COUNT REG
5148  023336  016761  155650  000004          MOV     $TMP5,LPR(R1)   ;LOAD LINE PARAMETERS
5149  023344  105062  032436                  CLRB    RBUF(R2)        ;INIT DATA BYTE IN RBUF TO START AT 000
5150  023350  110364  032436                  MOVB    R3,RBUF(R4)     ;SET CONSTANT IN HIGH BYTE
5151  023354  005211                          INC     (R1)            ;SELECT NEXT LINE
5152  023356  005203                          INC     R3              ;GENERATE NEW CONSTANT
5153  023360  062702  000002                  ADD     #2,R2           ;UPDATE POINTERS TO RBUF (EVEN/ODD)
5154  023364  062704  000002                  ADD     #2,R4
```

```
5155  023370  005367  155614                    DEC     $TMP4           ;COUNT ONE LINE SETUP
5156  023374  001352                            BNE     1$              ;BR TILL ALL 16. SET UP
5157  023376  012704  024660                    MOV     #MULPTB,R4      ;SET UP TABLE POINTER
5158  023402  016724  155606          2$:       MOV     $TMP6,(R4)+     ;SET UP BYTE COUNT ENTRY
5159  023406  022704  024720                    CMP     #MULPTB+40,R4   ;SET UP ALL COUNTS ?
5160  023412  001373                            BNE     2$              ;BR IF NOT
5161  023414  105011                            CLRB    (R1)            ;INIT SCR TO SELECT LINE 00
5162  023416  000207                            RTS     PC              ;RETURN TO PARITY TEST
5163
5164                                    ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
5165                                    ;TELETYPE
5166
5167  023420  104400                    INPARA: TYPE
5168  023422  032306                            VCWC                    ;"ASK FOR NO. WORDS BETWEEN VECTORS"
5169  023424  104407                            RDOCT                   ;READ OCTAL NO. FM TTY
5170  023426  012600                            MOV     (SP)+,R0        ;GET THE NO. HE TYPED
5171  023430  001412                            BEQ     3$              ;BR IF HE TYPED <CR>
5172  023432  022700  000004                    CMP     #4,R0           ;FOUR WORDS BETWEEN VECTORS ?
5173  023436  001404                            BEQ     2$              ;BR IF YES
5174  023440  022700  000010                    CMP     #10,R0          ;8. WORDS BETWEEN VECTORS ??
5175  023444  001404                            BEQ     3$              ;BR IF YES
5176  023446  000764                            BR      INPARA          ;ASK ALL OVER AGAIN
5177  023450  012700  000010          2$:       MOV     #10,R0          ;SET UP CONSTANT IN R0 FOUR 4 WORDS
5178  023454  000402                            BR      4$              ;CONTINUE
5179  023456  012700  000020          3$:       MOV     #20,R0          ;SET UP CONSTANT FOR 8. WORDS
5180  023462  000207                  4$:       RTS     PC              ;RETURN TO CALLER
5181
5182
5183
5184  023464  012700  177777          INPARC: MOV     #-1,R0          ;SET FLAG IN R0
5185  023470  000167  156452                    JMP     BEGINA          ;GO ASK FOR SELECT PARAMETER
5186
5187  023474  012767  177777  001374  INPARX: MOV     #-1,VCFLG       ;SET SETUP FLAG
5188  023502  000167  156440                    JMP     BEGINA          ;GO START UP
5189
5190  023506  104400                    INPAR:  TYPE                    ;ASK FOR DEVICE ADDRESS
5191  023510  031663                            INMSG1
5192  023512  104407                            RDOCT                   ;READ IN WHAT IS TYPED
5193  023514  012601                            MOV     (SP)+,R1        ;GET THE NO. HE TYPED
5194  023516  001403                            BEQ     INPAR1          ;BR IF DEFAULT
5195  023520  004767  000106                    JSR     PC,CHKADR       ;GO CHECK VALIDITY OF THE ADDR
5196  023524  000770                            BR      INPAR           ;ERROR BRANCH
5197
5198  023526  104400                    INPAR1: TYPE                    ;ASK FOR VECTOR ADDRESS
5199  023530  031727                            INMSG2
5200  023532  104407                            RDOCT                   ;READ IN WHAT HE TYPES
5201  023534  012601                            MOV     (SP)+,R1        ;GET THE ADDRESS
5202  023536  001403                            BEQ     INPAR3          ;BR IF DEFAULT
5203  023540  004767  000150                    JSR     PC,CHKVCT       ;GO CHECK VALIDITY OF VECTOR
5204  023544  000770                            BR      INPAR1          ;ERROR BRANCH
5205
5206  023546  104400                    INPAR3: TYPE                    ;ASK FOR DEVICE SELECTION PARAMETER
5207  023550  031776                            INMSG3
5208  023552  104407                            RDOCT                   ;READ IN WHAT HE TYPES
```

```
5209  023554  012601                          MOV     (SP)+,R1        ;GET THE SELECT PARAMETER
5210  023556  001402                          BEQ     INPAR4          ;BR IF  DEFAULT
5211  023560  010167  000622                  MOV     R1,DHSEL        ;SET UP DH11 SELECTION PARAMETER
5212
5213  023564  104400                  INPAR4: TYPE                    ;ASK FOR LINE SELECT PARAMETER
5214  023566  032174                          INMSG6
5215  023570  104407                          RDOCT                   ;GET WHAT HE TYPES
5216  023572  012601                          MOV     (SP)+,R1        ;GET PARAMETER
5217  023574  001403                          BEQ     1$              ;BR IF DEFAULT
5218  023576  010167  000606                  MOV     R1,LINSEL       ;SET UP LINE SELECT PARAMETER
5219  023602  000403                          BR      2$              ;CONTINUE
5220  023604  012767  177777  000576  1$:     MOV     #-1,LINSEL      ;SET UP DEFAULT (ALL LINES)
5221  023612  005777  155320          2$:     TST     @SWR            ;HALT AFTER SET UP ??
5222  023616  100003                          BPL     EXPAR           ;BR IF  NOT
5223  023620  104400                          TYPE                    ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
5224  023622  032236                          INMSG7
5225  023624  000000                          HALT                    ;DEPRESS CONTINUE TO RESUME TESTING
5226  023626  000167  156654          EXPAR:  JMP     START2          ;GO START UP THE PROGRAM
5227
5228
5229  023632  020127  160020          CHKADR: CMP     R1,#160020      ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
5230  023636  002001                          BGE     1$              ;BR IF YES
5231  023640  000422                          BR      4$              ;BR IF NOT
5232  023642  020127  160420          1$:     CMP     R1,#160420      ;IS IT BELOW THE HIGH LIMIT?
5233  023646  002401                          BLT     2$              ;BR IF YES
5234  023650  000416                          BR      4$              ;BR IF NOT
5235  023652  032701  000017          2$:     BIT     #17,R1          ;CORRECT BOUNDARY ?
5236  023656  001013                          BNE     4$              ;BR IF NOT
5237  023660  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
5238  023664  012702  024776                  MOV     #DHAOTB,R2      ;POINT TO BEGIN OF ADDR TABLE
5239  023670  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
5240  023672  062701  000020                  ADD     #20,R1          ;GENERATE NEXT DH11 ADDR
5241  023676  022702  025036                  CMP     #DHAOTB+40,R2   ;END OF TABLE ?
5242  023702  001372                          BNE     3$              ;BR IF NOT
5243  023704  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
5244  023706  104400                  4$:     TYPE                    ;TELL HIM HE GOOFED
5245  023710  032047                          INMSG4
5246  023712  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
5247
5248  023714  020127  000300          CHKVCT: CMP     R1,#300         ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
5249  023720  002001                          BGE     1$              ;BR IF YES
5250  023722  000421                          BR      4$              ;BR IF NOT
5251  023724  020127  001000          1$:     CMP     R1,#1000        ;IS IT BELOW THE HIGH LIMIT?
5252  023730  002401                          BLT     2$              ;BR IF YES
5253  023732  000415                          BR      4$              ;BR IF NOT
5254  023734  032701  000007          2$:     BIT     #7,R1           ;CORRECT BOUNDARY ?
5255  023740  001012                          BNE     4$              ;BR IF NOT
5256  023742  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN PC AROUND ERROR BRANCH
5257  023746  012702  025036                  MOV     #DHVCTB,R2      ;POINT TO BEGIN OF VECTOR TABLE
5258  023752  010122                  3$:     MOV     R1,(R2)+        ;SETR UP A TABLE ENTRY
5259  023754  060001                          ADD     R0,R1           ;GENERATE NEXT DH11 ADDR
5260  023756  022702  025076                  CMP     #DHVCTB+40,R2   ;END OF TABLE ?
5261  023762  001373                          BNE     3$              ;BR IF NOT
5262  023764  000402                          BR      5$              ;RETURN TO INPUT ROUTINES
```

```
5263  023766  104400                  4$:     TYPE                    ;TELL HIM HE GOOFED
5264  023770  032120                          INMSG5
5265  023772  000207                  5$:     RTS     PC              ;RETURN TO INPUT ROUTINES
5266
5267                                  ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
5268
5269  023774  012767  000340  155176  BUSER:  MOV     #340,$TMPO      ;SAVE THE PSW
5270  024002  010667  155166          MOV     SP,$REG6        ;SAVE THE SP
5271  024006  012601                  MOV     (SP)+,R1        ;GET THE TRAP PC
5272  024010  012602                  MOV     (SP)+,R2        ;GET THE TRAP PSW
5273  024012  116700  155064          MOVB    $TSTNM,R0       ;GET TEST NO.
5274  024016  012706  001100          MOV     #STACK,SP       ;RESET THE STACK POINTER
5275  024022  004767  176714          JSR     PC,SUER3        ;GO SET UP ERROR INFO
5276  024026  012767  024036  155054  MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
5277  024034  104027                  ERROR   27              ;UNEXPECTED BUS ERROR TRAP
5278  024036  000005          1$:     RESET                   ;PREPARE TO RESTART
5279  024040  004767  000240          JSR     PC,CHPS1        ;GO CLEAR PSW
5280  024044  000167  156532          JMP     REST1           ;GO RESTART THE PROGRAM
5281
5282  024050  012767  000340  155122  RESERR: MOV     #340,$TMPO      ;SAVE THE PSW
5283  024056  010667  155112          MOV     SP,$REG6        ;SAVE THE SP
5284  024062  012601                  MOV     (SP)+,R1        ;GET THE TRAP PC
5285  024064  012602                  MOV     (SP)+,R2        ;GET THE TRAP PSW
5286  024066  116700  155010          MOVB    $TSTNM,R0       ;GET TEST NO.
5287  024072  012706  001100          MOV     #STACK,SP       ;RESET THE STACK POINTER
5288  024076  004767  176640          JSR     PC,SUER3        ;GO SET UP ERROR INFO
5289  024102  012767  024112  155000  MOV     #1$,$LPERR      ;ALWAYS COME BACK TO 1$
5290  024110  104030                  ERROR   30              ;UNEXPECTED RSVD INSTR ERROR TRAP
5291  024112  000005          1$:     RESET                   ;PREPARE TO RESTART
5292  024114  004767  000164          JSR     PC,CHPS1        ;GO CLEAR PSW
5293  024120  000167  156456          JMP     REST1           ;GO RESTART THE PROGRAM
5294
5295                                  ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
5296                                  ;CATCHER IN THE DH11 VECTOR
5297
5298  024124  016703  000252  RESTRP: MOV     DHVCT,R3        ;GET VECTOR ADDRESS
5299  024130  010313                  MOV     R3,(R3)         ;RESTORE THE TRAP CATCHER
5300  024132  062723  000002          ADD     #2,(R3)+
5301  024136  005023                  CLR     (R3)+
5302  024140  010313                  MOV     R3,(R3)
5303  024142  062723  000002          ADD     #2,(R3)+
5304  024146  005023                  CLR     (R3)+
5305  024150  000207                  RTS     PC              ;RETURN TO CALLING TEST
5306
5307                                  ;THIS ROUTINE CALLED BY ANY TEST THAT NEEDS A TIMING WAIT LOOP
5308                                  ;"TIMEA" IS INITIALIZED BY THE CALLING ROUTINE TO THE MINIMUM REQUIRED
5309                                  ;VALUE AND "TIMEB" IS CLEARED TO 000000. IF A TIME OUT OCCURS THIS
5310                                  ;ROUTINE WILL MOVE THE RETURN PC AROUND THE "LOOP" BRANCH BACK IN
5311                                  ;THE ROUTINE THAT CALLED IT TO ALLOW REPORTING AN ERROR MESSAGE
5312
5313  024152  005267  001016  TIMEIT: INC     TIMEB           ;COUNT B
5314  024156  001005                  BNE     1$              ;BR IF NOT ZERO
5315  024160  005367  001006          DEC     TIMEA           ;COUNT TIME A
5316  024164  001002                  BNE     1$              ;BR IF NO TIMEOUT
```

```
5317  024166  062716  000002                    ADD     #2,(SP)         ;MOVE RETURN PC TO ALLOW ERROR REPORT
5318  024172  000207                    1$:     RTS     PC              ;RETURN TO THE CALLING TEST
5319
5320                                    ;THIS ROUTINE CALLED BY THE AUTO ECHO TEST TO SET UP FOR TRANSFERRING
5321                                    ;A BINARY COUNT TEST PATTERN ON ALL LINES
5322
5323  024174  012767  000020  155012    SETALL: MOV     #20,$TMP6       ;SET UP SIXTEEN LINES
5324  024202  005002                            CLR     R2              ;INIT A TABLE INDEX REG
5325  024204  012703  000200                    MOV     #200,R3         ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
5326  024210  012704  000001                    MOV     #1,R4           ;SET UP INDEX REG TO ODD BYTES
5327  024214  005011                            CLR     (R1)            ;START WITH LINE 00
5328  024216  012761  033436  000006    1$:     MOV     #TBUF,CAR(R1)   ;SET UP BUS ADDR REG
5329  024224  012761  177400  000010            MOV     #-400,BCR(R1)   ;SET UP BYTE COUNT REG
5330  024232  012761  031403  000004            MOV     #31403,LPR(R1)  ;SET UP FOR 4800 BAUD/8 BIT CHARS
5331  024240  105062  032436                    CLRB    RBUF(R2)        ;START WITH DATA CHAR OF 000
5332  024244  110364  032436                    MOVB    R3,RBUF(R4)     ;SET UP HIGH BYTE OF EXPECTED DATA
5333  024250  005211                            INC     (R1)            ;GEN NEW LINE NO. IN SCR
5334  024252  005203                            INC     R3              ;UPDATE THE POINTERS AND DATA
5335  024254  062702  000002                    ADD     #2,R2
5336  024260  062704  000002                    ADD     #2,R4
5337  024264  005367  154724                    DEC     $TMP6           ;COUNT ONE LINE DONE
5338  024270  001352                            BNE     1$              ;BR TIL ALL 16 SET UP
5339  024272  016767  000112  000356            MOV     LINSEL,LINACT   ;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
5340  024300  005011                            CLR     (R1)            ;PUT SCR REG BACK TO LINE 00
5341  024302  000207                            RTS     PC              ;RETURN TO AUTO ECHO TEST
5342
5343
5344                                    ;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
5345                                    ;TO BE LSI11 COMPATIBLE
5346
5347  024304  012746  000000            CHPS1:  MOV     #0,-(SP)        ;NEW PSW
5348  024310  012746  024316                    MOV     #1$,-(SP)       ;NEW PC
5349  024314  000002                            RTI                     ;CHANGE PSW
5350  024316  000207                    1$:     RTS     PC              ;RETURN TO CALLING TEST
5351
5352                                    ;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSW
5353                                    ;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTRS
5354
5355  024320  012746  000340            CHPS2:  MOV     #340,-(SP)      ;NEW PSW
5356  024324  012746  024332                    MOV     #1$,-(SP)       ;NEW PC
5357  024330  000002                            RTI                     ;CHANGE THE PSW
5358  024332  000207                    1$:     RTS     PC              ;RETURN TO CALLING TEST
5359
5360                                    ;THIS ROUTINE IS ALSO FOR LSI11 COMPATIBILITY AND IT IS CALLED
5361                                    ;TO SAVE THE PSW IN "$TMP0"
5362
5363  024334  005046                    SAPS:   CLR     -(SP)           ;TEMP STORAGE TO SAVE PSW
5364  024336  016746  153472                    MOV     34,-(SP)        ;SAVE TRAP VECTOR POINTER
5365  024342  012767  024352  153464            MOV     #1$,34          ;GO TO 1$ ON TRAP
5366  024350  104400                            TRAP                    ;GO TO IT
5367  024352  016666  000002  000006    1$:     MOV     2(SP),6(SP)     ;GET PSW SAVED
5368  024360  012716  024366                    MOV     #2$,(SP)        ;GO TO 2$ ON RTI
5369  024364  000002                            RTI
5370  024366  012667  153442            2$:     MOV     (SP)+,34        ;RESTORE VECTOR
```

```
 5371  024372  012667  154602                MOV     (SP)+,$TMP0     ;FINALLY SAVE PSW IN $TMP0
 5372  024376  000207                        RTS     PC
 5373
```

```
                        ;*****************************************************************
                        ;ADDITIONAL PROGRAM CONSTANTS AND VARIABLES
                        ;*****************************************************************
                        
                                NRC=2           ;INDEX CONST. TO ACCESS NEXT RCVD CHAR REG
                                LPR=4           ;INDEX CONST. TO ACCESS LINE PARAMETER REG.
                                CAR=6           ;INDEX CONST. TO ACCESS CURRENT ADDRESS REG.
                                BCR=10          ;INDEX CONST. TO ACCESS BYTE COUNT REG.
                                BAR=12          ;INDEX CONST. TO ACCESS BUFFER ACTIVE REG.
                                BKR=14          ;INDEX CONST. TO ACCESS BREAK CONTROL REG.
                                SSR=16          ;INDEX CONST. TO ACCESS SILO STATUS REG.
                        
024400  000000  DHADR:  0               ;HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
024402  000000  DHVCT:  0               ;HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
024404  000000  SELMSK: 0               ;BIT TST MARKER FOR SELECTING DH11'S
024406  000003  DHSEL:  3               ;SPECIFIES DH11'S SELECTED FOR TEST
024410  177777  LINSEL: 177777          ;SPECIFIES LINES TO TEST
024412  000000  LINMSK: 0               ;MARKER USED TO TEST FOR LINES TO TEST
024414  000000  LMSK1:  0               ;ALTERNATE MARKER TO SUPPORT THE
                                        ;SELECT LINES FEATURE
024416  000004  MSTCLR: .BLKW   4       ;FOUR WORD ADDRESS TABLE USED BY THE TEST THAT
                                        ;CHECKS OPERATION OF "MASTER CLR"
                        
024426  177777  PATRNA: 177777          ;BIT PATTERNS USED WITH "CAR" AND "BCR" TESTS
024430  125252          125252
024432  052525          052525
024434  000000          000000          ;TABLE TERMINATOR
                        
024436  000060  PATRNB: 60              ;BIT PATTERNS USED IN "CAR" MEM EXT BIT TEST
024440  000300          300
024442  000020          20
024444  000100          100
024446  000040          40
024450  000200          200
024452  000000          0               ;TABLE TERMINATOR
024454  000000          0               ;TABLE TERMINATOR
                        
                        ;THIS TABLE STORES THE BYTE COUNT AND LINE PARAMETERS FOR THE
                        ;8 SUBTESTS IN THE MULTILINE PARITY/DATA TEST
                        
024456  177400  PRTYTB: -400                   ;256 CHARS
024460  027363          27363                 ;2400 BAUD - ODD PARITY - 8 BITS
024462  177400          -400                  ;256 CHARS
024464  027323          27323                 ;2400 BAUD - EVEN PARITY - 8 BITS
024466  177600          -200                  ;128 CHARS
024470  027362          27362                 ;2400 BAUD - ODD PARITY - 7 BITS
024472  177600          -200                  ;128 CHARS
024474  027322          27322                 ;2400 BAUD - EVEN PARITY - 7 BITS
024476  177700          -100                  ;64 CHARS
024500  027361          27361                 ;2400 BAUD - ODD PARITY - 6 BITS
024502  177700          -100                  ;64 CHARS
024504  027321          27321                 ;2400 BAUD - EVEN PARITY - 6 BITS
024506  177740          -40                   ;32 CHARS
024510  027360          27360                 ;2400 BAUD - ODD PARITY - 5 BITS
```

```
                                        -40                     ;32 CHARS
                                        27320                   ;2400 BAUD - EVEN PARITY - 5 BITS


                                ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                ;TEST (ALL 1'S DATA TABLE)

                                AETAB:  100377          ;TEST DATA FOR LINE 00
                                        100777          ;TEST DATA FOR LINE 01
                                        101377
                                        101777
                                        102377
                                        102777
                                        103377
                                        103777
                                        104377
                                        104777
                                        105377
                                        105777
                                        106377
                                        106777
                                        107377
                                        107777          ;TEST DATA FOR LINE 17

                                ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
                                ;TEST (ALL 0'S DATA TABLE)

                                AETAB0: 100000          ;TEST DATA FOR LINE 00
                                        100400          ;TEST DATA FOR LINE 01
                                        101000
                                        101400
                                        102000
                                        102400
                                        103000
                                        103400
                                        104000
                                        104400
                                        105000
                                        105400
                                        106000
                                        106400
                                        107000
                                        107400          ;TEST DATA FOR LINE 17

                                ;THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
                                ;LINE IS DONE

                                LINBIT: BIT00                   ;DEACTIVATE LINE 00
                                        BIT01                   ;DEACTIVATE LINE 01
                                        BIT02
                                        BIT03
                                        BIT04
                                        BIT05
                                        BIT06
                                        BIT07
```

```
                                        BIT08
                                        BIT09
                                        BIT10
                                        BIT11
                                        BIT12
                                        BIT13
                                        BIT14
                                        BIT15                   ;DEACTIVATE LINE 17

                        LINACT: 0                               ;MAINTAINS STATUS OF ACTIVE LINES
                                                                ;DURING AUTO ECHO TEST 2


                        ;THIS TABLE CONTAINS 16. COUNTERS USED BYN THE MULTI-LINE
                        ;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED

                        MULPTB: .BLKW   16.                     ;SIXTEEN WORD COUNTERS TABLE

                        ;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
                        ;TEST

                        BRKTAB: 120000          ;TEST DATA FOR LINE 00
                                120400          ;TEST DATA FOR LINE 01
                                121000
                                121400
                                122000
                                122400
                                123000
                                123400
                                124000
                                124400
                                125000
                                125400
                                126000
                                126400
                                127000
                                127400          ;TEST DATA FOR LINE 17

                        RGMSK1: 131177          ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
                        RGMSK2: 46600           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
                        RGMSK3: 177767          ;MASK TO SPECIFY R/W BITS IN "LPR"
                        RGMSK4: 177777          ;MASK TO SPECIFY R/W BITS IN "BKR"
                        RGMSK5: 100077          ;MASK TO SPECIFY R/W BITS IN "SSR"
                        RGMSK6: 42200           ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
                        INTMSK: 30100           ;MASK USED TO SELECT INTR BITS TO TEST

                        ;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
                        ;SIXTEEN DH11'S

                        DHADTB: 160020          ;ADDRESS OF FIRST DH11
                                160040          ;ADDRESS OF SECOND DH11
                                160060
                                160100
                                160120
```

```
5537  025010  160140                          160140
5538  025012  160160                          160160
5539  025014  160200                          160200
5540  025016  160220                          160220
5541  025020  160240                          160240
5542  025022  160260                          160260
5543  025024  160300                          160300
5544  025026  160320                          160320
5545  025030  160340                          160340
5546  025032  160360                          160360
5547  025034  160400                          160400          ;ADDRESS OF THE LAST DH11
5548
5549                                  ;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
5550                                  ;TO SIXTEEN DH11'S
5551
5552  025036  000330                  DHVCTB: 330             ;ADDRESS OF VECTOR FOR FIRST DH11
5553  025040  000350                          350             ;ADDRESS OF VECTOR FOR SECOND DH11
5554  025042  000370                          370
5555  025044  000410                          410
5556  025046  000430                          430
5557  025050  000450                          450
5558  025052  000470                          470
5559  025054  000510                          510
5560  025056  000530                          530
5561  025060  000550                          550
5562  025062  000570                          570
5563  025064  000610                          610
5564  025066  000630                          630
5565  025070  000650                          650
5566  025072  000670                          670
5567  025074  000710                          710             ;ADDRESS OF VECTOR FOR LAST DH11
5568
5569  025076  000000                  VCFLG:  0               ;VECTOR SET UP FLAGG
5570
5571
5572                                  ;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
5573                                  ;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
5574                                  ;AND THE XMTTR LEVEL IN THE HIGH BYTE
5575
5576  025100  120240                  BRLVL:  120240          ;BRLEVELS FOR FIRST DH11
5577  025102  120240                          120240          ;BR LEVELS FOR SECOND DH11
5578  025104  120240                          120240
5579  025106  120240                          120240
5580  025110  120240                          120240
5581  025112  120240                          120240
5582  025114  120240                          120240
5583  025116  120240                          120240
5584  025120  120240                          120240
5585  025122  120240                          120240
5586  025124  120240                          120240
5587  025126  120240                          120240
5588  025130  120240                          120240
5589  025132  120240                          120240
5590  025134  120240                          120240
```

```
 5591  025136  120240                          120240          ;BR LEVELS FOR LAST DH11
 5592  025140     000          DHRLVL: .BYTE   0               ;BR LEVEL FOR RCVR
 5593  025141     000          DHTLVL: .BYTE   0               ;BR LEVEL FOR XMITTER
 5594  025142  000000          DHNUM:  0               ;CONTAINS NUMBER OF THE DH11 UNDER TEST
 5595  025144  000000          LINE:   0               ;CONTINES NUMBER OF THE LINE UNDER TEST
 5596  025146  000000          LINEA:  0               ;LOCATION TO SAVE LINE NUMBER
 5597                          ;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS
 5598
 5599  025150  000000          ADPTR:  0                       ;POINTS TO ADDRESS TABLE
 5600  025152  000000          VCPTR:  0                       ;POINTS TO VECTOR TABLE
 5601  025154  000000          BRPTR:  0                       ;POINTS TO BR LEVEL TABLE
 5602
 5603  025156  000000          TDATA1: 0                       ;DATA BUFFER FOR BASIC DATA TEST
 5604  025160  000037          TDATA2: 37                      ;TEST DATA FOR FIVE BIT CHAR
 5605  025162  000077                  77                      ;TEST DATA FOR SIX BIT CHAR
 5606  025164  000177                  177                     ;TEST DATA FOR SEVEN BIT CHAR
 5607  025166  000377                  377                     ;TEST DATA FOR EIGHT BIT CHAR
 5608  025170  000000          TITFLG: 0                       ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
 5609  025172  000000          TIMEA:  0                       ;GENERAL PURPOSE TIMERS
 5610  025174  000000          TIMEB:  0
 5611  025176  000000          TIMEC:  0                       ;TIMER FOR TIMING TESTS
 5612  025200  000000          TNULL:  0                       ;CONTAINS TWO NULL CHARS USED BY BREAK TEST
 5613
 5614
 5615
```

```
5617                                    ;*****************************************************************
5618                                    ;ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS
5619                                    ;*****************************************************************
5620
5621                                    ;INFORMATION FOR MESSAGE 1
5622
5623  025202 044104 030461 051040  EM1:    .ASCIZ  'DH11 REGISTER REFERENCE CAUSED TIMEOUT'
      025210 043505 051511 042524
      025216 020122 042522 042506
      025224 042522 041516 020105
      025232 040503 051525 042105
      025240 052040 046511 007505
      025246 052123 000000
5624  025251 020040 050050 024503  DH1:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR'
      025257 020040 020040 050050
      025264 024523 020040 020040
      025272 051450 024520 020040
      025300 020040 042524 052123
      025306 020040 042040 053105
      025314 042101 020122 051040
      025322 043505 042101 000122
5625                                    .EVEN
5626  025330 001116 001200 001174  DT1:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,0
      025336 001160 001162 001164
      025344 000000
5627
5628                                    ;INFORMATION FOR MESSAGE 2
5629
5630  025346 054523 052123 046505  EM2:    .ASCIZ  'SYSTEM CONTROL REGISTER ERROR'
      025354 041440 047117 051124
      025362 046117 051040 043505
      025370 051511 042524 020122
      025376 051105 047522 000122
5631  025404 024040 041520 020051  DH2:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  REGADR   WAS     S/B'
      025412 020040 024040 051520
      025420 020051 020040 024040
      025426 050123 020051 020040
      025434 052040 051505 020124
      025442 020040 042504 040526
      025450 051104 020040 042522
      025456 040507 051104 020040
      025464 053440 051501 020040
      025472 020040 051440 041057
      025500 000
5632                    025502          .EVEN
5633  025502 001116 001200 001174  DT2:    .WORD   $ERRPC,$TMPO,$REG6,$REG0,$REG1,$REG2,$REG3,$REG4,0
      025510 001160 001162 001164
      025516 001166 001170 000000
5634
5635                                    ;INFORMATION FOR MESSAGE 3
5636
5637  025524 044104 030461 046440  EM3:    .ASCIZ  'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'
      025532 051501 042524 020122
      025540 046103 040505 020122
```

```
                025546  040506  046111  042105
                025554  052040  020117  046103
                025562  020122  050123  041505
                025570  043111  042511  020104
                025576  042522  000107
 5638
 5639                                           ;INFORMATION FOR MESSAGE 4
 5640
 5641   025602  044514  042516  050040  EM4:    .ASCIZ  'LINE PARAMETER REGISTER ERROR'
        025610  051101  046501  052105
        025616  051105  051040  043505
        025624  051511  042524  020122
        025632  051105  047522  000122
 5642
 5643                                           ;INFORMATION FOR MESSAGE 5
 5644
 5645   025640  051102  040505  020113  EM5:    .ASCIZ  'BREAK CONTROL REGISTER ERROR'
        025646  047503  052116  047522
        025654  020114  042522  044507
        025662  052123  051105  042440
        025670  051122  051117     000
 5646
 5647                                           ;INFORMATION FOR MESSAGE 6
 5648
 5649   025675     123  046111  020117  EM6:    .ASCIZ  'SILO STATUS REGISTER ERROR'
        025702  052123  052101  051525
        025710  051040  043505  051511
        025716  042524  020122  051105
        025724  047522  000122
 5650
 5651                                           ;INFORMATION FOR MESSAGE 7
 5652
 5653   025730  052503  051122  047105  EM7:    .ASCIZ  'CURRENT ADDRESS REGISTER ERROR - LINE #XX'
        025736  020124  042101  051104
        025744  051505  020123  042522
        025752  044507  052123  051105
        025760  042440  051122  051117
        025766  026440  046040  047111
        025774  020105  054043  000130
 5654
 5655                                           ;INFORMATION FOR MESSAGE 10
 5656
 5657   026002  054502  042524  041440  EM10:   .ASCIZ  'BYTE COUNTER REGISTER ERROR - LINE #XX'
        026010  052517  052116  051105
        026016  051040  043505  051511
        026024  042524  020122  051105
        026032  047522  020122  020055
        026040  044514  042516  021440
        026046  054130     000
 5658
 5659                                           ;INFORMATION FOR MESSAGE 11
 5660
 5661   026051     125  042516  050130  EM11:   .ASCIZ  'UNEXPECTED DH11 RCVR INTERRUPT'
        026056  041505  042524  020104
```

```
        026064  044104  030461  051040
        026072  053103  020122  047111
        026100  042524  051122  050125
        026106  000124
  5662
  5663                                  ;INFORMATION FOR MESSAGE 12
  5664
  5665  026110  047125  054105  042520  EM12:   .ASCIZ  'UNEXPECTED DH11 XMITTR INTERRUPT'
        026116  052103  042105  042040
        026124  030510  020061  046530
        026132  052111  051124  044440
        026140  052116  051105  052522
        026146  052120     000
  5666
  5667                                  ;INFORMATION FOR MESSAGE 13
  5668
  5669  026151     103  040510  020122  EM13:   .ASCIZ  'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'
        026156  053101  044501  040514
        026164  046102  020105  040506
        026172  046111  042105  052040
        026200  020117  042507  042516
        026206  040522  042524  051040
        026214  053103  020122  047111
        026222  042524  051122  050125
        026230  000124
  5670
  5671                                  ;INFORMATION FOR MESSAGE 14
  5672
  5673  026232  051124  047101  046523  EM14:   .ASCIZ  'TRANSMITTER NPR LOGIC ERROR - LINE #  '
        026240  052111  042524  020122
        026246  050116  020122  047514
        026254  044507  020103  051105
        026262  047522  020122  020055
        026270  044514  042516  021440
        026276  020040     000
  5674
  5675                                  ;INFORMATION FOR MESSAGE 15
  5676
  5677  026301     130  044515  052124  EM15:   .ASCIZ  'XMITTR FAILED TO INTERRUPT - LINE #  '
        026306  020122  040506  046111
        026314  042105  052040  020117
        026322  047111  042524  051122
        026330  050125  020124  020055
        026336  044514  042516  021440
        026344  020040     000
  5678
  5679                                  ;INFORMATION FOR MESSAGE 16
  5680
  5681  026347     122  053103  020122  EM16:   .ASCIZ  'RCVR FAILED TO INTERRUPT'
        026354  040506  046111  042105
        026362  052040  020117  047111
        026370  042524  051122  050125
        026376  000124
  5682
```

```
5683                                        ;INFORMATION FOR MESSAGE 17
5684
5685  026400  051124  047101  046523  EM17:   .ASCIZ  'TRANSMITTER TIMING ERROR - LINE #  '
      026406  052111  042524  020122
      026414  044524  044515  043516
      026422  042440  051122  051117
      026430  026440  046040  047111
      026436  020105  020043  000040
5686  026444  024040  041520  020051  DH6:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  SPEED   TIMEB   TIMEC'
      026452  020040  024040  051520
      026460  020051  020040  024040
      026466  050123  020051  020040
      026474  052040  051505  020124
      026502  020040  042504  040524
      026510  051104  020040  050123
      026516  042505  020104  020040
      026524  044524  042515  020102
      026532  020040  044524  042515
      026540  000103
5687
5688                                        ;INFORMATION FOR MESSAGE 20
5689
5690  026542  042522  042503  053111  EM20:   .ASCIZ  'RECEIVER TIMING ERROR - LINE #  '
      026550  051105  052040  046511
      026556  047111  020107  051105
      026564  047522  020122  020055
      026572  044514  042516  021440
      026600  020040     000
5691
5692                                        ;INFORMATION FOR MESSAGE 21
5693
5694  026603     122  053103  020122  EM21:   .ASCIZ  'RCVR FAILED TO INTERRUPT - LINE #  '
      026610  040506  046111  042105
      026616  052040  020117  047111
      026624  042524  051122  050125
      026632  020124  020055  044514
      026640  042516  021440  020040
      026646     000
5695
5696                                        ;INFORMATION FOR MESSAGE 22
5697
5698  026647     103  040510  020122  EM22:   .ASCIZ  'CHAR AVAIL FAILED TO SET ON TIME - LINE #  '
      026654  053101  044501  020114
      026662  040506  046111  042105
      026670  052040  020117  042523
      026676  020124  047117  052040
      026704  046511  020105  020055
      026712  044514  042516  021440
      026720  020040     000
5699
5700                                        ;INFORMATION FOR MESSAGE 23
5701
5702  026723     102  051501  041511  EM23:   .ASCIZ  'BASIC DATA TEST ERROR - LINE #  '
      026730  042040  052101  020101
```

```
                026736  042524  052123  042440
                026744  051122  051117  026440
                026752  046040  047111  020105
                026760  020043  000040
 5703           026764  024040  041520  020051  DH7:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  WAS     S/B'
                026772  020040  024040  051520
                027000  020051  024040  024040
                027006  050123  020051  020040
                027014  052040  051505  020124
                027022  020040  042504  040526
                027030  051104  020040  044103
                027036  046122  043516  020040
                027044  053440  051501  020040
                027052  020040  051440  041057
                027060     000
 5704
 5705                                           ;INFORMATION FOR MESSAGE 24
 5706
 5707           027061     101  052125  020117  EM24:   .ASCIZ  'AUTO ECHO TEST ERROR - LINE #  '
                027066  041505  047510  052040
                027074  051505  020124  051105
                027102  047522  020122  020055
                027110  044514  042516  021440
                027116  020040     000
 5708
 5709                                           ;INFORMATION FOR MESSAGE 25
 5710
 5711           027121     102  042522  045501  EM25:   .ASCIZ  'BREAK BIT TEST ERROR - LINE #  '
                027126  041040  052111  052040
                027134  051505  020124  051105
                027142  047522  020122  020055
                027150  044514  042516  021440
                027156  020040     000
 5712
 5713                                           ;INFORMATION FOR MESSAGE 26
 5714
 5715           027161     110  046101  026506  EM26:   .ASCIZ  'HALF-DUPLEX TEST ERROR - LINE #  '
                027166  052504  046120  054105
                027174  052040  051505  020124
                027202  051105  047522  020122
                027210  020055  044514  042516
                027216  021440  020040     000
 5716
 5717                                           ;INFORMATION FOR MESSAGE 27
 5718
 5719           027223     125  042516  050130  EM27:   .ASCIZ  'UNEXPECTED BUS ERROR TRAP'
                027230  041505  042524  020104
                027236  052502  020123  051105
                027244  047522  020122  051124
                027252  050101     000
 5720           027255     040  050050  024503  DH3:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST   TRPPC   TRPPS '
                027262  020040  020040  050050
                027270  024523  020040  020040
                027276  051450  024520  020040
```

```
        027304  020040  042524  052123
        027312  020040  052040  050122
        027320  041520  020040  052040
        027326  050122  051520  000040
 5721                                   .EVEN
 5722   027334  001116  001200  001174  DT3:    .WORD   SERRPC,$TMP0,$REG6,$REG0,$REG1,$REG2,0
        027342  001160  001162  001164
        027350  000000
 5723
 5724                                   ;INFORMATION    FOR MESSAGE 30
 5725
 5726   027352  047125  054105  042520  EM30:   .ASCIZ  'UNEXPECTED RSVD INSTR TRAP'
        027360  052103  042105  051040
        027366  053123  020104  047111
        027374  052123  020122  051124
        027402  050101     000
 5727
 5728                                   ;INFORMATION FOR MESSAGE 31
 5729
 5730   027405     101  052125  020117  EM31:   .ASCIZ  'AUTO ECHO DATA COMPARE ERROR - LINE #  '
        027412  041505  047510  042040
        027420  052101  020101  047503
        027426  050115  051101  020105
        027434  051105  047522  020122
        027442  020055  044514  042516
        027450  021440  020040     000
 5731   027455     040  050050  024503  DH4:    .ASCIZ  ' (PC)    (PS)    (SP)    TEST    WASADR  SBADR   WAS     S/B'
        027462  020040  020040  050050
        027470  024523  020040  020040
        027476  051450  024520  020040
        027504  020040  042524  052123
        027512  020040  053440  051501
        027520  042101  020122  051440
        027526  040502  051104  020040
        027534  020040  040527  020123
        027542  020040  020040  027523
        027550  000102
 5732
 5733                                   ;INFORMATION FOR MESSAGE 32
 5734
 5735   027552  052501  047524  042440  EM32:   .ASCIZ  'AUTO ECHO TEST TIMEOUT - LINE #  '
        027560  044103  020117  042524
        027566  052123  052040  046511
        027574  047505  052125  026440
        027602  046040  047111  020105
        027610  020043  000040
 5736   027614  024040  041520  020051  DH5:    .ASCIZ  ' (PC)   (LPRG)   TEST'
        027622  020040  046050  051120
        027630  024507  020040  052040
        027636  051505  000124
 5737                                   .EVEN
 5738   027642  001116  001200  001204  DT4:    .WORD   SERRPC,$TMP0,$TMP2,0
        027650  000000
 5739
```

```
5740                                            ;INFORMATION FOR MESSAGE 33
5741
5742  027652 040520 044522 054524  EM33:   .ASCIZ  'PARITY LOGIC TEST ERROR - LINE #  '
      027660 046040 043517 041511
      027666 052040 051505 020124
      027674 051105 047522 020122
      027702 020055 044514 042516
      027710 021440 020040    000
5743
5744                                            ;INFORMATION FOR MESSAGE 34
5745
5746  027715    115 046125 044524  EM34:   .ASCIZ  'MULTI-LINE PARITY DATA TEST ERROR - LINE #   - SUBTEST #  '
      027723 046055 047111 020105
      027731 040520 044522 054524
      027737 042040 052101 020101
      027745 042524 052123 042440
      027753 051122 051117 026440
      027761 046040 047111 020105
      027767 020043 020040 020055
      027775 052523 052122 051505
      030003 020124 020043 000040
5747
5748                                            ;INFORMATION FOR MESSAGE 35
5749
5750  030010 052515 052114 026511  EM35:   .ASCIZ  'MULTI-LINE PARITY DATA TEST TIMEOUT'
      030016 044514 042516 050040
      030024 051101 052111 020131
      030032 040504 040524 052040
      030040 051505 020124 044524
      030046 042515 052517 000124
5751  030054 024040 041520 020051  DH14:   .ASCIZ  ' (PC)   (LPRG)  ACTLIN'
      030062 020040 046050 051120
      030070 024507 020040 041501
      030076 046124 047111    000
5752                  030104            .EVEN
5753  030104 001116 001200 001206  DT6:    .WORD   SERRPC,STMP0,STMP3,0
      030112 000000
5754
5755                                            ;INFORMATION FOR MESSAGE 36
5756
5757  030114 044103 051101 040440  EM36:   .ASCIZ  'CHAR AVAILABLE TIMEOUT'
      030122 040526 046111 041101
      030130 042514 052040 046511
      030136 047505 052125    000
5758
5759                                            ;INFORMATION FOR MESSAGE 37
5760
5761  030143    104 052101 020101  EM37:   .ASCIZ  'DATA COMPARE ERROR - LINE #  '
      030151 047503 050115 051101
      030157 020105 051105 047522
      030165 020122 020055 044514
      030173 042516 021440 020040
      030201    000
5762
```

```
                                        ;INFORMATION FOR MESSAGE 40
                                        
030201    102  043125  042506  EM40:   .ASCIZ  'BUFFER ACTIVE REG ERROR - LINE #  '
030206  020122  041501  044524
030214  042526  051040  043505
030222  042440  051122  051117
030230  026440  046040  047111
030236  020105  020043  000040

                                        ;INFORMATION FOR MESSAGE 41

030244  041522  051126  043040  EM41:   .ASCIZ  'RCVR FALSE INTERRUPT'
030252  046101  042523  044440
030260  052116  051105  052522
030266  052120     000

                                        ;INFORMATION FOR MESSAGE 42

030271    123  046111  020117  EM42:   .ASCIZ  'SILO OVERFLOW ERROR'
030276  053117  051105  046106
030304  053517  042440  051122
030312  051117     000

                                        ;INFORMATION FOR MESSAGE 43

030315    123  046111  020117  EM43:   .ASCIZ  'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
030322  053117  051105  046106
030330  053517  043040  044501
030336  042514  020104  047524
030344  043440  047105  051105
030352  052101  020105  041522
030360  051126  044440  052116
030366  051105  052522  052120
030374     000

                                        ;INFORMATION FOR MESSAGE 44

030375    116  047117  042440  EM44:   .ASCIZ  'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
030402  020130  042515  047515
030410  054522  043040  044501
030416  042514  020104  047524
030424  043440  047105  051105
030432  052101  020105  046530
030440  052111  051124  044440
030446  052116  051105  052522
030454  052120     000

                                        ;INFORMATION FOR MESSAGE 45

030457    130  044515  020124  EM45:   .ASCIZ  'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
030464  047504  042516  043040
030472  044501  042514  020104
030500  047524  043440  047105
030506  051105  052101  020105
```

```
                030514  046530  052111  051124
                030522  044440  052116  051105
                030530  052522  052120     000
 5786
 5787                                           ;INFORMATION FOR MESSAGE 46
 5788
 5789   030535     103  051125  042522  EM46:   .ASCIZ  'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE #  '
        03 542  052116  040440  042104
        0  550  042522  051523  046440
        0  556  046505  051117  020131
        0  564  040520  052124  051105
        0  572  051516  052040  051505
        0  600  020124  051105  047522
        0  606  020122  020055  044514
        0  614  042516  021440  020040
        0  622     000
 5790   0  623     040  050050  024503  DH10:   .ASCIZ  ' (PC)   LINEWR  PATTRN   TEST   DEVADR  REGADR  WAS     S/B'
        0  630  020040  046040  047111
        0  636  053505  020122  050040
        0  644  052101  051124  020116
        0  652  020040  042524  052123
        0  660  020040  042040  053105
        0  666  042101  020122  051040
        0  674  043505  042101  020122
        0  702  020040  040527  020123
        030710  020040  020040  027523
        030716  000102
 5791                                           .EVEN
 5792   030720  001116  001200  001202  DTS:    .WORD   $ERRPC,$TMP0,$TMP1,$REG0,$REG1,$REG2,$REG3,$REG4,0
        030726  001160  001162  001164
        030734  001166  001170  000000
 5793
 5794                                           ;INFORMATION FOR MESSAGE 47
 5795
 5796   030742  054502  042524  041440  EM47:   .ASCIZ  'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE #  '
        03 750  052517  052116  046440
        03 756  046505  051117  020131
        03 764  040520  052124  051105
        030772  051516  052040  051505
        031000  020124  051105  047522
        031006  020122  020055  044514
        031014  042516  021440  020040
        031022     000
 5797                                           ;INFORMATION FOR MESSAGE 50
 5798
 5799   031023     124  051505  020124  EM50:   .ASCIZ  'TEST TIMEOUT WAITING FOR XMIT DONE - LINE #  '
        031030  044524  042515  052517
        031036  020124  040527  052111
        031044  047111  020107  047506
        031052  020122  046530  052111
        031060  042040  047117  020105
        031066  020055  044514  042516
        031074  021440  020040     000
 5800
```

```
5801                                        ;INFORMATION FOR MESSAGE 51
5802
5803  031101    116  051120  046040  EMS1:   .ASCIZ  'NPR LOGIC TEST 2 ERROR'
      031106 043517  041511  052040
      031114 051505  020124  020062
      031122 051105  047522  000122
5804  031130 024040  041520  020051  DH11:   .ASCIZ  ' (PC)   LINACT  LINCHK   TEST   DEVADR  REGADR   WAS     S/B'
      031136 020040  044514  040516
      031144 052103  020040  044514
      031152 041516  045510  020040
      031160 052040  051505  020124
      031166 020040  042504  040526
      031174 051104  020040  042522
      031202 040507  051104  020040
      031210 053440  051501  020040
      031216 020040  051440  041057
      031224    000
5805
5806                                        ;INFORMATION FOR MESSAGE 52
5807
5808  031225    102  051501  041511  EMS2:   .ASCIZ  'BASIC DATA COMPARE ERROR'
      031232 042040  052101  020101
      031240 047503  050115  051101
      031246 020105  051105  047522
      031254 000122
5809
5810                                        ;INFORMATION FOR MESSAGE 53
5811
5812  031256 024040  041520  020051  DH12:   .ASCIZ  ' (PC)   SPEED    (SP)    TEST   DEVADR  REGADR   WAS     S/B'
      031264 020040  050123  042505
      031272 020104  020040  024040
      031300 050123  020051  020040
      031306 052040  051505  020124
      031314 020040  042504  040526
      031322 051104  020040  042522
      031330 040507  051104  020040
      031336 053440  051501  020040
      031344 020040  051440  041057
      031352    000
5813
5814                                        ;INFORMATION FOR MESSAGE 55
5815
5816  031353    040  050050  024503  DH13:   .ASCIZ  ' (PC)    (PS)    (SP)    TEST   DEVADR  CHRLNG  SCRWAS  SCRS/B'
      031360 020040  020040  050050
      031366 024523  020040  020040
      031374 051450  024520  020040
      031402 020040  042524  052123
      031410 020040  042040  053105
      031416 042101  020122  041440
      031424 051110  047114  020107
      031432 051440  051103  040527
      031440 020123  051440  051103
      031446 027523  000102
5817
```

```
                                        ;INFORMATIOPN FOR MESSAGE 56
                                        ;
031452 053117 051105 052522 EM56:   .ASCIZ  'OVERRUN BIT FAILED TO SET - LINE #  '
031460 020116 044502 020124
031466 040506 046111 042105
031474 052040 020117 042523
031502 020124 020055 044514
031510 042516 021440 020040
031516     000
                                        ;
                                        ;INFORMATION FOR MESSAGE 57
                                        ;
031517     123 047524 040522 EM57:   .ASCIZ  'STORAGE OVERFLOW BIT FAILED - LINE #  '
031524 042507 047440 042526
031532 043122 047514 020127
031540 044502 020124 040506
031546 046111 042105 026440
031554 046040 047111 020105
031562 020043 000040
                                        ;
                                        .EVEN
                                        ;MISCELLANEOUS MESSAGES
                                        ;
031566 005015 040515 047111 TITLE:  .ASCIZ  <15><12>'MAINDEC-11-DZDHM-A  DH11 DIAGNOSTIC'<15><12>
031574 042504 026503 030461
031602 042055 042132 046510
031610 040455 020040 044104
031616 030461 042040 040511
031624 047107 051517 044524
031632 006503 000012
031636 005015 042524 052123 TITLE2: .ASCIZ  <15><12>'TESTING DH11 #  '<15><12>
031644 047111 020107 044104
031652 030461 021440 020040
031660 005015     000
031663     015 052012 050131 INMSG1: .ASCIZ  <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>
031670 020105 041523 020122
031676 042101 051104 051505
031704 020123 047506 020122
031712 044506 051522 020124
031720 044104 030461 005015
031726     000
031727     015 052012 050131 INMSG2: .ASCIZ  <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
031734 020105 042526 052103
031742 051117 040440 042104
031750 042522 051523 043040
031756 051117 043040 051111
031764 042123 042040 030510
031772 030461 000012
031776 005015 054524 042520 INMSG3: .ASCIZ  <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
032004 040040 030510 020061
032012 042504 044526 042503
032020 051440 046105 041505
032026 044524 047117 050040
032034 051101 046501 052105
032042 051105 005015     000
```

```
 5833  032047  000015  044412  053116  INMSG4: .ASCIZ  <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
       032054  046101  042111  042040
       032062  030510  020061  041523
       032070  020122  042101  051104
       032076  051505  020123  020055
       032104  051124  020131  043501
       032112  044501  006516  000012
 5834  032120  005015  047111  040526  INMSG5: .ASCIZ  <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
       032126  044514  020104  044104
       032134  030461  053040  041505
       032142  047524  020122  042101
       032150  051104  051505  020123
       032156  020055  051124  020131
       032164  043501  044501  006516
       032172  000012
 5835  032174  005015  054524  042520  INMSG6: .ASCIZ  <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
       032202  046040  047111  020105
       032210  042523  042514  052103
       032216  047511  020116  040520
       032224  040522  042515  042524
       032232  006522  000012
 5836  032236  005015  042504  051120  INMSG7: .ASCIZ  <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
       032244  051505  020123  041442
       032252  047117  044524  052516
       032260  021105  052040  020117
       032266  052123  051101  020124
       032274  042524  052123  047111
       032302  006507  000012
 5837
 5838  032306  005015  054524  042520  VCWC:   .ASCIZ  <15><12>'TYPE NO. OF WORDS (OCTAL) BETWEEN VECTORS (4 OR 10)'<15><12>
       032314  047040  027117  047440
       032322  020106  047527  042122
       032330  020123  047450  052103
       032336  046101  020051  042502
       032344  053524  042505  020116
       032352  042526  052103  051117
       032360  020123  032050  047440
       032366  020122  030061  006451
       032374  000012
 5839                                  .EVEN
 5840                                  ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3
 5841
 5842  032376  000020                  RCNT:   .BLKW   16.
 5843
 5844                                  ;256. WORD RECEIVER INPUT BUFFER
 5845
 5846  032436  000400                  RBUF:   .BLKW   256.
 5847
 5848
 5849                                  ;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
 5850
 5851                                  .EVEN
 5852  033436  000400                  TBUF:   .BLKB   256.
 5853
```

```
 5854          000001                    .END
```

```
ABASE = 000000          2259
ACDMI = 000000          2259
ACDM2 = 000000          2259
ACPUOP= 000000          2259
ADDMO = 000000          2259
ADDMI = 000000          2259
ADDMIO= 000000          2259
ADDMII= 000000          2259
ADDMI2= 000000          2259
ADDMI3= 000000          2259
ADDMI4= 000000          2259
ADDMI5= 000000          2259
ADDM2 = 000000          2259
ADDM3 = 000000          2259
ADDM4 = 000000          2259
ADDM5 = 000000          2259
ADDM6 = 000000          2259
ADDM7 = 000000          2259
ADDM8 = 000000          2259
ADDM9 = 000000          2259
ADEVCT= 000000          2259
ADEVM = 000000          2259
ADPTR   025150          2619#   2626#   2634    4975#   5599#
AENV  = 000000          2259
AENVM = 000000          2259
AETAB   024516          4520    4597    4694    4728    4732    4733    5435#
AETABD  024556          4589    5455#
AFATAL= 000000          2259
AMAORI= 000000          2259
AMAOR2= 000000          2259
AMAOR3= 000000          2259
AMAOR4= 000000          2259
AMAMSI= 000000          2259
AMAMS2= 000000          2259
AMAMS3= 000000          2259
AMAMS4= 000000          2259
AMSGAD= 000000          2259
AMSGLG= 000000          2259
AMSGTY= 000000          2259
AMTYPI= 000000          2259
AMTYP2= 000000          2259
AMTYP3= 000000          2259
AMTYP4= 000000          2259
APASS = 000000          2259
APRIOR= 000000          2259
APTCSU= 000040          4990    4991#
APTENV= 000001          4986    4990    4991#
APTSIZ= 000200          2596    4991#
APTSPC= 000100          4990    4991#
ASUREG= 000000          2259
ATESTM= 000000          2259
AUNIT = 000000          2259
AUSWR = 000000          2259
AVECTI= 000000          2259
```

```
AVECT2= 000000  2259
BAR   = 000012  3644* 3675  3679  3735* 3996* 4065* 4130* 4196* 4253* 4306  4368* 4429* 4490
                4496  4540* 4622* 4707* 4774* 4806* 4810  4819  4879* 4883  4890  4892  4927*
                5383*
BCR   = 000010  3075  3155  3235  3326  3642* 3698  3703  3760  3771  3994* 4062* 4127* 4192*
                4249* 4320  4365* 4534* 4617* 4697* 4772* 4803* 4877* 4925* 5120* 5132* 5147*
                5329* 5[illegible]*
BEGIN   002144  2248  2[illegible]*
BEGINA  002146  2594* 5185  5188
BIT0  = 000001  2258*
BIT00 = 000001  2258*  5475
BIT01 = 000002  2258*  5476
BIT02 = 000004  2258*  5477
BIT03 = 000010  2258*  5478
BIT04 = 000020  2258*  5479
BIT05 = 000040  2258*  5480
BIT06 = 000100  2258*  3478  3807  5481
BIT07 = 000200  2258*  3479  3657  4081  4147  5482
BIT08 = 000400  2258*  4985  5483
BIT09 = 001000  2258*  2767  2769  2780  2781  2792  3477  3517  3557  3597  4252  4985  4986
                5484
BIT1  = 000002  2258*
BIT10 = 002000  2258*  3559  5485
BIT11 = 004000  2258*  2688  3439  3474  3514  3554  3594  3640  3709  3730  3805  3858  3891
                3935  3990  4060  4125  4190  4246  4358  4419  4445  4452  4466  4497  4511
                4519  4611  4692  4767  4800  4873  4922  4985  5486
BIT12 = 010000  2258*  3[illegible]9  5487
BIT13 = 020000  2258*  3[illegible]8  3598  3645  4986  5488
BIT14 = 040000  2258*  3518  4000  4006  4022  4449  4953  4985  5489
BIT15 = 100000  2258*  3599  3747  3806  3861  3865  3896  3899  3939  3943  4083  4147  4554
                4583  4682  4745  4907  4940  4953  5490
BIT2  = 000004  2258*
BIT3  = 000010  2258*
BIT4  = 000020  2258*
BIT5  = 000040  2258*
BIT6  = 000100  2258*  3981
BIT7  = 000200  2258*
BIT8  = 000400  2258*
BIT9  = 001000  2258*
BKR   = 000014  2679  2843  2898* 2952  3014* 3397* 4367* 4805* 5384*
BPTVEC= 000014  2258*
BRKTAB  024720  4753  5504*
BRLVL   025100  2621  5576*
BRPTR   025154  2621* 2628* 2636  4977* 5601*
BUSER   023774  2599  5269*
CAR   = 000006  3029  3121  3189  3281  3381* 3391* 3398  3686  3692  3776  3787  3993* 4063*
                4128* 4193* 4248* 4334  4366* 4535* 4536* 4615* 4616* 4696* 4771* 4802* 4876*
                4924* 5119* 5146* 5328* 5381*
CHKACR  023632  5195  5[illegible]*
CHPVCT  023714  5203  5[illegible]*
CHPS1   024304  3442  3480  3520  3560  3600  3646  3715  3808  3844  4432  4436  5279  5292
                5347*
CHPS2   024320  3438  3476  3516  3556  3596  3639  3803  4423  5355*
CLCABC  023176  3638  3731  4532  4768  4874  5117*
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
DDISP = 177570          2258#  2259
DHADR   024400          2634#  4245  4304  4415  4610  4985  5084#  5387#
DHADTB  024776          2619   5238  5241  5532#
DHNUM   025142          2622*  2625*  2638  4974*  5594#
DHRLVL  025140          2636*  3422  3470  3510  3550  3590  3802  4426  5592#
DHSEL   024406          2629   4980  5211*  5390#
DHTLVL  025141          2439   3473  3513  3553  3593  3630  5593#
DHVCT   024402          2635*  3424  3468  3508  3548  3588  3627  3710  3800  3846  4424  5298  5388#
DHVCTB  025036          2620   5237  5260  5552#
DHI     025251          2262   5624#
DHI0    030623          2324   5631  5790#
DHI1    031130          2545   5804#
DHI2    031256          2553   5666  5812#
DHI3    031353          2573   5816#
DHI4    030054          2461   5751#
DH2     025404          2270   5577  2284  2291  2298  2305  2312  2319  2326  2333  2340  2347  2354
                        2375   5582  2397  2404  2411  2447  2482  2503  2510  2517  2538  2552  2580
                        2587   5631#
DH3     027255          2418   5656  5720#
DH4     027455          2432   5704  2475  5731#
DH5     027614          2439   5709  2469  2496  5736#
DH6     026444          2361   5686  5686#
DH7     026764          2389   5703#
DISPLA  001140          2259#  2536*  4985*  4986#
DISPRE  000174          2247#  2536
DSWR  = 177570          2258#  2259
DT1     025330          2264   5626#
DT2     025502          2271   2277  2285  2292  2299  2306  2313  2320  2327  2334  2341  2348  2355
                        2382   2389  2376  2383  2390  2398  2405  2412  2433  2448  2455  2476  2483
                        2504   2511  2518  2539  2553  2560  2567  2574  2581  2588  5633#
DT3     027334          2419   5646  5722#
DT4     027642          2440   5709  2490  2497  5738#
DT5     030720          2543   5652  2546  5792#
DT6     030104          2462   5753#
EMTVEC= 000030          2258#  2536#
EM1     025202          2262   5623#
EM10    026002          2311   3105  3162  4330  5657#
EM11    026051          2318   5661#
EM12    026110          2325   5665#
EM13    026151          2332   5669#
EM14    026232          2339   3721  5673#
EM15    026301          2346   3661  5677#
EM16    026347          2353   5681#
EM17    026400          2360   4107  5685#
EM2     025346          2269   5630#
EM20    026542          2367   4171  5690#
EM21    026603          2374   5694#
EM22    026647          2381   2565  2572  4152  4214  4272  4386  5698#
EM23    026723          2318   4229  5702#
EM24    027061          2356   4510  4576  5707#
EM25    027121          2403   4756  4825  4844  4861  5711#
EM26    027161          2410   4897  4912  5715#
EM27    027223          2417   5719#
EM3     025524          2276   5637#
```

```
EM30    027352          2424    5726#
EM31    027405          2431    4658    4673    4738    5730#
EM32    027552          2438    4635    5735#
EM33    027653          2446    4400    5742#
EM34    027715          2453    4473    4476    5746#
EM35    030010          2460    5750#
EM36    030114          2467    5757#
EM37    030143          2474    4295    5761#
EM4     025602          2283    5641#
EM40    030201          2481    4316    5765#
EM41    030244          2488    5769#
EM42    030271          2495    5773#
EM43    030315          2502    5777#
EM44    030375          2509    5781#
EM45    030457          2516    5785#
EM46    030535          2523    3222    3313    5789#
EM47    030742          2530    3268    3358    5796#
EM5     025640          2290    5645#
EM50    031023          2537    2558    3753    4088    4945    5799#
EM51    03:101          2544    5803#
EM52    031225          2551    5808#
EM56    031452          2579    4966    5820#
EM57    031517          2586    4013    4033    4047    5824#
EM6     025675          2297    5649#
EM7     025730          2304    3059    3128    3410    4344    5653#
ENDA    017674          4921    4972#   4981
ERRVEC= 000004          2258#   2599#   2600#   2652    2653#   2668#   4985#
EXPAR   023626          5222    5226#
GNS     = ******  U     2247    4994
INMSG1  031663          5191    5830#
INMSG2  031727          5199    5831#
INMSG3  031776          5207    5832#
INMSG4  032047          5245    5833#
INMSG5  032120          5264    5834#
INMSG6  032174          5214    5835#
INMSG7  032236          5224    5836#
INPAR   023506          2617    5190#   5196
INPARA  023420          2610    5167#   5176
INPARC  023464          2248    5184#
INPARX  023474          2247    5187#
INP  1  023526          5194    5198#   5204
INP  3  023546          2616    5202#   5206#
INP  4  023564          2210    5214#
INTMSK  024774          3431    5827#
IOTVEC= 000020          2258*   2596#
LDBAR   023240          3732    5130#
LDTBF1  022610          2642    5005#
LINACT  024656          4428#   4463#   4504    4510    4621#   4679#   4680    4706#   4746#   5339#   5492#
LINBIT  024616          4483    4679    4746    5475#
LINE    025144          3034    3038    3046    3050    3058    3066#   3080    3084    3092    3096    3104    3112#   3125
                        3127    3159    3161    3195    3213    3241    3259    3236    3304    3331    3349    3388    3401
                        3637    3641    3660    3720    3733    3748    3752    3761    3768    3777    3784    3992    4007
                        4012    4027    4032    4046    4061    4084    4087    4106    4126    4148    4151    4170    4191
                        4210    4213    4220    4228    4247    4268    4271    4285    4294    4315    4329    4343    4360
```

```
                        4362  4381  4385  4399  4533  4556  4559  4575  4588* 4614  4634  4644  4672
                        4769  4794  4801  4824  4843  4860  4875  4896  4908  4911  4923  4939  4944
                        4950  4965  4985* 5070  5072* 5074* 5076* 5083* 5595*
LINEA   025146          3195* 3204  3241* 3250  3286* 3295  3331* 3340  5051  5596*
LINMSK  024412          3644  3735  3996  4065  4130  4196  4253  4367  4368  4540  4621  4774  4805
                        4806  4879  4927  5073* 5077* 5079  5392*
LINSEL  024410          3030  3031* 3069* 3076  3077* 3115* 4422  4428  4429  4528  4605  4622  4706
                        4707  4761  5079  5218* 5220* 5339  5391*
LMSK1   024414          4521* 4525* 4528  4598* 4602* 4605  4754* 4758* 4761  5393*
LPR   = 000004          2680  2794* 2809  2887* 2914  3002* 3390* 3643* 3734* 3995* 4064* 4129* 4194*
                        4195* 4250* 4364* 4421  4539* 4551* 4570* 4583* 4618* 4632  4682* 4698* 4717
                        4745* 4773* 4804* 4878* 4926* 5148* 5330* 5380*
MSTCLR  024416          2673  2685  5395*
MULPTB  024660          4481* 5157  5159  5499*
NRC   = 000002          3821  3827  3905  4017  4018  4218  4276  4390  4396  4456  4468  4565  4572
                        4626  4721  4734  4850  4857  4954  4962  5379*
PATRNA  024426          3190  3236  5398*
PATRNB  024436          3372  5403*
PC    =%000007          2258* 2610* 2642* 2662* 2692* 2718* 2727* 2749* 2776* 2788* 2800* 2825* 2834*
                        2889* 2898* 2894* 2905* 2932* 2943* 2970* 2981* 3009* 3021* 3032* 3044* 3056*
                        3078* 3090* 3102* 3122* 3136* 3146* 3156* 3170* 3180* 3193* 3219* 3239* 3265*
                        3284* 3310* 3329* 3355* 3386* 3407* 3438* 3442* 3446* 3448* 3453* 3455* 3461*
                        3476* 3480* 3483* 3488* 3492* 3497* 3501* 3516* 3520* 3523* 3528* 3532* 3537*
                        3541* 3556* 3560* 3563* 3568* 3572* 3577* 3591* 3596* 3600* 3603* 3608* 3612*
                        3617* 3621* 3631* 3638* 3639* 3646* 3651* 3655* 3658* 3669* 3670* 3671* 3678*
                        3681* 3692* 3690* 3693* 3694* 3701* 3704* 3705* 3715* 3722* 3728* 3731* 3732*
                        3741* 3745* 3750* 3772* 3788* 3803* 3808* 3813* 3816* 3826* 3829* 3838* 3840*
                        3844* 3876* 3918* 3950* 3951* 3952* 3959* 3962* 3963* 3973* 3983* 4002* 4005*
                        4010* 4025* 4030* 4040* 4044* 4056* 4071* 4085* 4093* 4104* 4121* 4136* 4149*
                        4162* 4168* 4185* 4202* 4206* 4211* 4226* 4241* 4259* 4263* 4269* 4292* 4310*
                        4313* 4324* 4327* 4338* 4341* 4356* 4374* 4378* 4383* 4394* 4397* 4420* 4423*
                        4432* 4436* 4437* 4465* 4470* 4492* 4506* 4530* 4532* 4546* 4550* 4557* 4569*
                        4573* 4607* 4612* 4628* 4650* 4655* 4666* 4670* 4713* 4731* 4735* 4763* 4768*
                        4780* 4794* 4792* 4812* 4816* 4822* 4833* 4841* 4855* 4858* 4871* 4874* 4885*
                        4889* 4894* 4904* 4909* 4920* 4933* 4936* 4942* 4960* 4963* 4984* 4986* 4987*
                        4990* 4991* 4995  5011* 5016* 5023* 5028* 5037* 5045* 5058* 5085* 5125* 5137*
                        5162* 5180* 5195* 5203* 5246* 5265* 5275* 5279* 5288* 5292* 5305* 5318* 5341*
                        5350* 5358* 5372*
PIRQ  = 177772          2258*
PIRQVE= 000240          2258*
PSTAT   177756          4408  4412  5415*
R0    = 000000          2258*
R1    = 000040          2258*
R2    = 000100          2258*
R3    = 000140          2258*
R4    = 000200          2258*
R5    = 000240          2258*
R6    = 000300          2258*
R7    = 000340          2258*
PS    = 177776          2258*
PSW   = 177776          2258*
PWRVEC= 000024          2258*  2596* 4995*
RBUF    032436          4251  4277  4281  4301  4462  4467  4469  4480* 4647  4652  4653  4677* 5149*
                        5150* 5331* 5332* 5846*
```

```
RCNT    032376          4695    4727*   4742    5842#
RDCHR = 104405          4992    4994#
RDLIN = 104406          4993    4994#
RDOCT = 104407          4994#   5169    5192    5200    5208    5215
RESERR  024050          2601    5282#
RESTRP  024124          3461    3501    3541    3581    3621    4437    5298#
RESTRT  002544          2625#   2633
RESTI   002602          2631#   5280    5293
RESVEC= 000010          2258#   2601*   2602*
RGMSK1  024760          2704    5521#
RGMSK2  024762          2736    5522#
RGMSK3  024764          2811    2916    2922    2926    5523#
RGMSK4  024766          2845    2954    2960    2963    5524#
RGMSK5  024770          2879    2932    2948    3001    5525#
RGMSK6  024772          2760    5526#
RSTRTA  002612          2630    2634#   4983    4995
R0      =%000000        2258#   2593*   2612    2614    4984*   4987*   4989*   4990*   4991*   4993*   4994*   4995*   5017*
                        5018    5029*   5030    5042    5052*   5053    5097    5098*   5100*   5101    5108*   5109*   5110
                        5111*   5170*   5172    5174    5177*   5179*   5184*   5259    5273*   5286*
R1      =%000001        2258#   2649    2674    2675    2676    2677    2688*   2701    2712*   2722*   2734    2744*   2758
                        2770*   2782*   2794*   2808    2819*   2829*   2842    2853*   2863*   2876    2887*   2898*   2913
                        2924*   2937*   2951    2964*   2975*   2989    3002*   3014*   3028    3038*   3039*   3050*   3051*
                        3074    3084*   3095*   3096*   3097*   3120    3125*   3130*   3141*   3154    3159*   3164*   3175*
                        3188    3198*   3204*   3205*   3210*   3211*   3234    3244*   3250*   3251*   3256*   3257*   3280
                        3289*   3295*   3296*   3301*   3302*   3325    3334*   3340*   3341*   3346*   3347*   3370    3378*
                        3379*   3380*   3381*   3388*   3389*   3390*   3391*   3395*   3396*   3397*   3398    3423    3439*
                        3441*   3447    3454    3467    3474*   3477*   3478*   3479*   3484    3485*   3486*   3493    3495*
                        3496*   3507    3514*   3517*   3518*   3519*   3524    3525*   3526*   3533    3534*   3536*   3547
                        3554*   3557*   3558*   3559*   3564    3565*   3566*   3573    3575*   3576*   3587    3594*   3597*
                        3598*   3599*   3604    3605*   3606*   3613    3615*   3616*   3635    3640*   3641*   3642*   3643*
                        3644*   3645*   3656    3665    3669    3675    3686    3698    3709*   3730*   3733*   3734*   3735*
                        3739    3746    3749    3758*   3760    3770    3776    3786    3805*   3806*   3807*   3814    3840
                        3821    3845*   3855    3858*   3888    3891*   3905    3932    3935*   3945    3990*   3991    3992*
                        3993*   3994*   3995*   3996*   4000    4008    4017    4018    4022    4028    4037    4043    4060*
                        4061*   4062*   4063*   4064*   4065*   4069    4080    4082    4125*   4126*   4127*   4129*   4129*
                        4130*   4134    4145    4146    4190*   4191*   4192*   4193*   4194*   4195*   4196*   4200    4207
                        4218    4245*   4246*   4247*   4248*   4249*   4250*   4252*   4253*   4257    4264    4276    4281*
                        4298    4300*   4301    4304*   4305    4319    4333    4358*   4362*   4364*   4365*   4366*   4367*
                        4368*   4372    4379    4382    4390    4395    4415*   4419*   4421    4427*   4429*   4442    4445*
                        4449    4452*   4456    4466*   4468*   4490    4496    4497*   4511*   4519*   4533*   4534*   4535*
                        4536*   4539*   4540*   4544    4551*   4552    4553    4565    4570*   4571    4583*   4610*   4611*
                        4614*   4615*   4616*   4617*   4618*   4622*   4626    4632    4654*   4669*   4682*   4683    4692*
                        4696*   4697*   4698*   4701*   4707*   4711    4717    4721    4734*   4744*   4745*   4767*   4769*
                        4771*   4772*   4773*   4774*   4778    4786    4800*   4801*   4802*   4803*   4804*   4805*   4806*
                        4810    4818    4830    4835    4850    4856    4873*   4875*   4876*   4877*   4878*   4879*   4883
                        4890    4891    4901    4905    4906    4922*   4923*   4924*   4925*   4926*   4927*   4931    4937
                        4941    4954    4961    4985*   4989*   4991*   4993*   4995*   5005*   5007*   5019    5031    5043
                        5054    5097    5099*   5107*   5110*   5111*   5118*   5119*   5120*   5124*   5131*   5132*   5136*
                        5142*   5146*   5147*   5148*   5151*   5161*   5193*   5201*   5209*   5211    5216*   5218    5229
                        5232    5235    5239    5240*   5248    5251    5254    5258    5259*   5271*   5284*   5327*   5328*
                        5329*   5330*   5333*   5340*
R2      =%000002        2258#   2649*   2650    2654*   2656*   2657    2658    2684*   2687*   2689    2701*   2711*   2713*
                        2714    2723*   2724    2734*   2743*   2745*   2746    2758*   2768*   2769*   2771*   2772    2780*
                        2783*   2784    2793*   2795*   2796    2808*   2809*   2818*   2820*   2821    2830*   2831    2842*
```

```
                    2843*  2852*  2854*  2855   2864*  2865   2876*  2877*  2886*  2888*  2889   2899*  2900
                    2913*  2914*  2923*  2926*  2927*  2928   2938*  2939   2951*  2952*  2962*  2963*  2965*
                    2966   2976*  2977   2989*  2990*  3000*  3001*  3003*  3004   3015*  3016   3028*  3029*
                    3040*  3052   3074*  3075*  3086*  3098   3120*  3121*  3131*  3132   3142*  3143   3154*
                    3155*  3165*  3166   3176*  3177   3188*  3189*  3199*  3206*  3212   3234*  3235*  3245*
                    3252*  3258   3280*  3281*  3290*  3297*  3303   3325*  3326*  3335*  3342*  3348   3370*
                    3371*  3399   3423*  3467*  3507*  3547*  3587*  3635*  3679*  3692*  3703*  3749*  3770*
                    3771*  3786*  3787*  3827*  3839*  3855*  3856*  3861*  3865*  3870   3888*  3889*  3896*
                    3899*  3911   3932*  3933*  3937*  3939*  3943*  3967   3991*  4042*  4062*  4099*  4100*
                    4101*  4102*  4103*  4146*  4163*  4164*  4165*  4166*  4167*  4188*  4195   4232*  4233
                    4264*  4265   4280*  4282   4299*  4305*  4306*  4307   4311   4319*  4320*  4321   4325
                    4333*  4334*  4335   4339   4382*  4395*  4396*  4424*  4425*  4426*  4457*  4458*  4459*
                    4460   4461*  4462   4467   4469*  4480*  4481*  4483   4552*  4571*  4572*  4642*  4643*
                    4647   4650   4653*  4667*  4677*  4679   4694*  4696   4700*  4722*  4723*  4724*  4725
                    4726*  4727*  4728   4732   4733*  4742   4746   4786*  4787*  4788   4818*  4819*  4820
                    4835*  4836*  4837   4856*  4857*  4891*  4892*  4905*  4941*  4961*  4962*  4989*  4993*
                    4995*  5006*  5007   5008*  5009   5020   5032   5044   5055   5097   5101*  5102*  5103*
                    5104*  5105*  5106*  5107   5111*  5143*  5149*  5153*  5238*  5239*  5241   5257*  5258*
                    5260   5272*  5285*  5324*  5331*  5335*
R3      =%000003    2258*  2668*  2714*  2715   2724*  2746*  2772*  2773   2784*  2785   2796*  2797   2821*
                    2822   2831*  2855*  2856   2865*  2889*  2890*  2891   2900*  2901*  2902   2928*  2929
                    2939*  2940   2966*  2967   2977*  2978   3004*  3005*  3006   3016*  3017*  3018   3052*
                    3053   3098*  3099   3132*  3133   3143*  3166*  3167   3177*  3212*  3216   3258*  3262
                    3303*  3307   3348*  3352   3398*  3399*  3400*  3404   3424*  3425*  3426*  3427   3428*
                    3429*  3447*  3454*  3468*  3469*  3470*  3471   3472*  3473*  3484*  3493*  3508*  3509*
                    3510*  3511   3512*  3513*  3524*  3533*  3548*  3549*  3550*  3551   3552*  3553*  3564*
                    3573*  3588*  3589*  3590*  3591   3592*  3593*  3604*  3613*  3627*  3628*  3629*  3630*
                    3656*  3657*  3669*  3675*  3686*  3687   3698*  3710*  3711*  3712*  3713*  3714*  3746*
                    3760*  3764   3776*  3780   3800*  3801*  3802*  3809*  3810*  3814*  3821*  3823   3832*
                    3833*  3835   3846*  3847*  3848*  3849*  3862*  3863*  3870*  3871*  3873   3894*  3900*
                    3903*  3908*  3911*  3912*  3915   3940*  3941*  3967*  4008*  4009*  4028*  4029*  4043*
                    4080*  4081*  4092*  4094   4145*  4156*  4158   4207*  4208*  4218*  4223   4265*  4266*
                    4288*  4289   4311*  4325*  4339*  4379*  4390*  4391   4456*  4457   4462   4553*  4554*
                    4565*  4566   4626*  4639   4647   4663   4695*  4699*  4721*  4722   4728   4788*  4789*
                    4820*  4837*  4838*  4850*  4851   4890*  4906*  4907*  4937*  4938*  4954*  4957   4988*
                    4989*  4992*  4995*  5033   5056   5144*  5150   5152*  5298*  5299*  5300*  5301*  5302*
                    5303*  5304*  5325*  5332   5334*
R4      =%000004    2258*  2682*  2710*  2715   2721*  2742*  2766*  2767*  2773   2781*  2785   2792*  2793
                    2797   2817*  2822   2828*  2851*  2856   2862*  2885*  2891   2897*  2902   2922*  2925*
                    2929   2936*  2940   2960*  2961*  2967   2974*  2978   2998*  2999*  3006   3013*  3018
                    3036*  3040   3041*  3048*  3053   3063*  3082*  3086   3087*  3094*  3099   3109*  3124*
                    3131   3133   3140*  3142   3158*  3165   3167   3174*  3176   3209*  3215*  3216   3255*
                    3261*  3262   3300*  3306*  3307   3345*  3351*  3352   3394*  3403*  3404   3440*  3487*
                    3494*  3527*  3534*  3567*  3574*  3607*  3614*  3636*  3637*  3680*  3691*  3702*  3747*
                    3748*  3759*  3763*  3764   3775*  3779*  3780   3815*  3822*  3823   3834*  3835   3860*
                    3866*  3872*  3873   3895*  3897*  3904*  3906*  3913*  3914*  3915   3968*  3969*  3970*
                    3971*  3972*  4006*  4007*  4026*  4027*  4041*  4083*  4084*  4093*  4094   4147*  4148*
                    4157*  4158   4209*  4210*  4219*  4220*  4221*  4222*  4223   4267*  4268*  4282*  4283*
                    4284*  4285*  4286*  4207*  4299   4312*  4326*  4340*  4359*  4360*  4361*  4380*  4381*
                    4391   4467*  4527*  4555*  4556*  4566   4639*  4640*  4641*  4642   4644   4651   4652*
                    4668*  4732*  4760*  4785   4790*  4791*  4796*  4817   4821*  4826*  4834   4839*  4840*
                    4845*  4851   4893*  4908*  4939*  4940*  4950*  4951*  4952*  4953*  4957   4988*  4995*
                    5034   5057   5145*  5150*  5154*  5157*  5158*  5159   5326*  5332*  5336*
R5      =%000005    2258*  2637*  2650*  2651*  2658   2673*  2674*  2675*  2676*  2677*  2678*  2679*  2680*
```

```
SW0    = 000001          2258#
SW00   = 000001          2258#
SW01   = 000002          2258#
SW02   = 000004          2258#
SW03   = 000010          2258#
SW04   = 000020          2258#
SW05   = 000040          2258#
SW06   = 000100          2258#
SW07   = 000200          2258#
SW08   = 000400          2258#
SW09   = 001000          2258#
SW1    = 000002          2258#
SW10   = 002000          2258#
SW11   = 004000          2258#
SW12   = 010000          2258#
SW13   = 020000          2258#
SW14   = 040000          2258#
SW15   = 100000          2258#
SW2    = 000004          2258#
SW3    = 000010          2258#
SW4    = 000020          2258#
SW5    = 000040          2258#
SW6    = 000100          2258#
SW7    = 000200          2258#
SW8    = 000400          2258#
SW9    = 001000          2258#
TBITVE= 000014           2258#
TBUF    033436           4248    4280    4335    4340    4802    4876    4924    5005    5146    5328    5852#
TCRLF  = 000200          4990#
TDATA1  025156           4363#   4366    5603#
TDATA2  025160           4187    5604#
THT    = 000011          4990#
TIMEA   025172           3648#   3737#   3998#   4067#   4132#   4198#   4255#   4370#   4488#   4502#   4542#   4624#   4709#
                         4776#   4809#   4891#   4929#   5315#   5609#
TIMEB   025174           3649#   3738#   3998#   4068#   4092    4110    4133#   4156    4174    4199#   4256#   4371#   4489#
                         4503#   4543#   4625#   4710#   4777#   4809#   4882#   4930#   5313#   5610#
TIMEC   025176           4059#   4093    4110#   4124#   4157    4174#   5611#
TIMEIT  024152           3651    3741    4002    4071    4136    4202    4259    4374    4492    4506    4546    4628    4713
                         4780    4812    4865    4933    5313#
TITFLG  025170           2594#   2603    2607#   5608#
TITLE   031566           2606    5608#
TITLE2  031636           2639    2641    5829#
TKVEC  = 000060          2258#
TNULL   025200           4771    5612#
TPVEC  = 000064          2258#
TRAPVE= 000034           2258#   2596#
TRTVEC= 000014           2258#
TST1    002662           2647#
TST10   003734           2848    2873#
TST11   004070           2882    2910#
TST12   004226           2919    2948#
TST13   004364           2957    2985#
TST14   004532           2995    3026#
TST15   004730           3072#
```

```
TST16   005126          3118#
TST17   005260          3123    3152#
TST2    002760          2671#
TST20   005412          3157    3186#
TST21   005626          3192    3232#
TST22   006042          3238    3278#
TST23   006250          3285    3323#
TST24   006456          3330    3368#
TST25   006724          3375    3420#
TST26   007116          3464#
TST27   007304          3504#
TST3    002906          2686    2698#
TST30   007472          3544#
TST31   007660          3584#
TST32   010046          3624#
TST33   010506          3717    3725#
TST34   011042          3729    3797#
TST35   011302          3853#
TST36   011432          3886#
TST37   011614          3929#
TST4    003204          2707    2732#
TST40   012036          3985#
TST41   012354          3989    4053#
TST42   012670          4057    4118#
TST43   013200          4122    4182#
TST44   013446          4186    4238#
TST45   014126          4353#
TST46   014366          4357    4405#
TST47   015100          4438    4515#
TST5    003274          2739    2755#
TST50   015460          4587    4594#
TST51   016126          4608    4689#
TST52   016434          4719    4740    4750#
TST53   017202          4868#
TST54   017424          4872    4917#
TST6    003470          2763    2805#
TST7    003612          2814    2839#
TYPOS = 104404          4984    4994#
TYPE  = 104400          2605    2640    4984    4986    4987    4988    4989    4990    4992    4993    4994#   4995    5167
                        5190    5198    5206    5213    5223    5244    5263
TYPOC = 104401          4987    4994#
TYPON = 104403          4994#
TYPOS = 104402          4994#
VCFLG   025076          2608    2611#   5187#   5569#
VCPTR   025152          2620#   2627#   2635    4976#   5600#
VCWC    032306          5168    5838#
$APTHD  000214          2252#
$ASTAT= ****** U        4991
$ATYC   021646          4991#
$ATY1   021622          4991#
$ATY3   021630          4990    4991#
$ATY4   021640          4986    4991#
$BASE   001304          2259#
$BOARD  001122          2259#
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
$BDDAT  001126            2259#   2596
$CDW1   001310            2259#
$CDW2   001312            2259#
$CHARC  021616            4990#*
$CMTAG  001100            2259#   2596
$CM1  = 000010            2259#
$CM2  = 000020            2259#
$CM3  = 000010            2259#
$CMN  = 000010            2259#
$CPUOP  001256            2259#
$CRLF   001225            2259#   4986    4987    4990    4992    4993
$DBLK   021334            4989#
$DDW0   001314            2259#
$DDW1   001316            2259#
$DDW10  001340            2259#
$DDW11  001342            2259#
$DDW12  001344            2259#
$DDW13  001346            2259#
$DDW14  001350            2259#
$DDW15  001352            2259#
$DDW2   001320            2259#
$DDW3   001322            2259#
$DDW4   001324            2259#
$DDW5   001326            2259#
$DDW6   001330            2259#
$DDW7   001[illegible]    2259#
$DDW8   001[illegible]    2259#
$DDW9   001[illegible]    2259#
$DEVCT  001240            2259#
$DEVM   001306            2259#
$DOAGN  020056            4974#
$DTBL   021324            4979#
$ENDAD  020046            2251    4984#
$ENDCT  020014            4984#
$ENDMG  020062            4994#
$ENULL  020077            4984#
$ENV    001250            2259#   4986    4990    4991
$ENVM   001251            2259#   2596    4990    4991
$EOP    017760            4973#   4979    4984#
$EOPCT  020006            4984#
$ERFLG  001103            2259#   4985#   4986#
$ERMAX  001115            2259#   2596#   4985#
$ERROR  020370            2596    4986#
$ERRPC  001116            2259#   4[illegible]6#  4987    5626    5633    5722    5738    5753    5792
$ERRTB  001354            2259#   4987
$ERRTY  020536            4986    4987#
$ERTTL  001112            2259#   4986#
$ESCAP  001222            2259#   2596#   4985#   4986
$ETABL  001250            2259#
$ETEND  001354            2252    2259#
$FATAL  001232            2259#   4991#
$FFLG   022066            4991#*
$FILLC  001154            2259#   4990
$FILLS  001153            2259#   4990
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
$GDADR  001120          2259#
$GDDAT  001124          2259#
$GET42  020036          4984#
$HO   = 000001          2244
$HIBTS  000214          2252#
$HIOCT  022400          4993#*
$ICNT   001104          2259#   4985#
$ILLUP  022572          4995#
$ITEMB  001114          2259#   4986#   4987
$LF     001226          2259#   4986    4990    4992    4993
$LFLG   022065          4991#*
$LOOP   001106          2259#   2596#   2644#   4985#
$LPLOR  001110          2259#   2596#   2664#   2693#   2700#   2750#   2757#   2807#   2841#   2875#   2912#   2950#   2998#
                        3060#   3106#   3137#   3147#   3171#   3181#   3223#   3269#   3314#   3359#   3411#   3422#   3466#
                        3506#   3546#   3586#   3626#   3727#   3799#   3877#   3919#   3931#   3987#   4055#   4120#   4184#
                        4240#   4355#   4407#   4517#   4596#   4691#   4752#   4870#   4919#   4985#   4986    5276#   5289#
$MADR1  001262          2259#
$MADR2  001266          2259#
$MADR3  001272          2259#
$MADR4  001276          2259#
$MAIL   001230          2252    2259#   2596    2647    2671    2698    2732    2755    2805    2839    2873    2910    2948
                        2986    3026    3072    3118    3152    3186    3232    3278    3323    3368    3420    3464    3504
                        3544    3584    3624    3725    3797    3853    3886    3929    3985    4053    4118    4182    4238
                        4353    4405    4515    4594    4689    4750    4868    4917    4985    4986    4990
$MAMS1  001260          2259#
$MAMS2  001264          2259#
$MAMS3  001270          2259#
$MAMS4  001274          2259#
$MBADR  000216          2252#
$MFLG   022064          4991#*
$MSGAD  001244          2259#   4991#
$MSGLG  001246          2259#   4991#
$MSGTY  001230          2259#   4991#
$MTYP1  001261          2259#
$MTYP2  001265          2259#
$MTYP3  001271          2259#
$MTYP4  001275          2259#
$MXCNT  020366          4985#
$NULL   001152          2259#   4990
$NWTST= 000001          2647#   2671#   2698#   2732#   2755#   2805#   2839#   2873#   2910#   2948#   2986#   3026#   3072#
                        3118#   3152#   3186#   3232#   3278#   3323#   3368#   3420#   3464#   3504#   3544#   3584#   3624#
                        3725#   3797#   3853#   3886#   3929#   3985#   4053#   4118#   4182#   4238#   4353#   4405#   4515#
                        4594#   4689#   4750#   4868#   4917#
$OCNT   021114          4988#*
$OMODE  021116          4988#*
$OVER   020352          4985#
$PASS   001236          2259#   2596#   4984#   4985
$PASTM  000222          2252#
$POWER  022600          4995#
$PRIOR  001302          2259#
$PWRAD  022566          4995#
$PWRON  022444          2596    4995#
$PWRMG  022562          4995#
$PWRUP  022512          4995#
```

```
$QUES   001224          2259#  4986   4990   4992   4993
$ROCHR  022070          4992#  4994
$RODEC= ****** U        4994
$ROLIN  022124          4992#  4994
$ROOCT  022242          4993#  4994
$ROSZ = 000010          4992#
$REGAD  001156          2259#
$REG0   001160          2259#  5018#  5030#  5042#  5053#  5626   5633   5722   5792
$REG1   001162          2259#  5019#  5031#  5043#  5054#  5626   5633   5722   5792
$REG2   001164          2259#  5020#  5032#  5044#  5055#  5626   5633   5722   5792
$REG3   001166          2259#  5033#  5056#  5633   5792
$REG4   001170          2259#  5034#  5057#  5633   5792
$REG5   001172          2259#
$REG6   001174          2259#  5021#  5022#  5035#  5036#  5270#  5283#  5626   5633   5722
$REG7   001176          2259#
$R2A  = ****** U        4994
$SAVRE= ****** U        4994
$SAVR6  022576          4995#*
$SCOPE  020102          2596   4985#
$SETUP= 000017          2595#  2596   4984   4986
$STUP = 177777          2595#
$SVLAD  020316          4985#
$SVPC = 000214          2251#
$SWR  = 165400          2237#  2244   2245   2259   2596   2647   2671   2698   2732   2755   2805   2839   2873
                        2910   2948   2986   3026   3072   3118   3152   3186   3232   3278   3323   3368   3420
                        3464   3504   3544   3584   3624   3725   3797   3853   3916   3929   3975   4053   4118
                        4182   4238   4353   4405   4515   4594   4689   4750   4868   4917   4964   4985   4986
                        4995
$SWREG  001252          2259#  2595
$SWRMK= 000000          2245   4985
STESTN  001234          2259#  2647#  2671#  2698#  2732#  2755#  2805#  2839#  2873#  2910#  2948#  2986#  3026#
                        3072#  3118#  3152#  3186#  3232#  3278#  3323#  3368#  3420#  3464#  3504#  3544#  3584#
                        3624#  3725#  3797#  3853#  3916#  3929#  3975#  4053#  4118#  4182#  4238#  4353#  4405#
                        4515#  4594#  4689#  4750#  4868#  4917#  4985#
STIMES  001220          2259#  2596#  4984#  4985#
$TKB    001144          2259#  4992
$TKS    001142          2259#  4992
STMP0   001200          2259#  3767#  3768#  3783#  3784#  4075#  4076#  4077#  4078#  4079#  4140#  4141#  4142#
                        4143#  4144#  4421#  4632#  4717#  5060#  5061#  5269#  5282#  5371#  5626   5633   5722
                        5738   5753   5792
STMP1   001202          2259#  3191#  3206   3209   3237#  3252   3255   3283#  3328#  3373#  3389   3769#  3785#
                        4785#  4796   4817#  4826   4834#  4845   4949#  4955#  5792
STMP2   001204          2259#  3208#  3210   3213   3221   3225#  3227   3254#  3256   3259   3267   3272#  3273
                        3299#  3301   3304   3312   3317#  3318   3344#  3346   3349   3357   3362#  3363   3374#
                        3394   4430#  4431#  4619#  4620#  4704#  4705#  5738
STMP3   001206          2259#  3197#  3198   3200#  3201   3243#  3244   3246#  3247   3288#  3289   3291#  3292
                        3333#  3334   3336#  3337   3377#  3379   3382#  3383   4422#  4496#  4510#  4518#  4586#
                        5753
STMP4   001210          2259#  3393#  3396   3401   3409   3414#  3415   4460#  4472   5141#  5155#
STMP5   001212          2259#  4417#  4651#  4657   5148
STMP6   001214          2259#  4416#  4558#  4564#  4725#  4737   4744   5147   5158   5323#  5337#
STMP7   001216          2259#  3070#  3069   3076#  3115   3757#  3758   3761   3769   3777   3785   3791#  3792
                        3857#  3879   3880#  3881   3890#  3893   3903   3922#  3923   3934#  3936   3968   3972
                        3975   3980#  3981   4017#  4018#  4058#  4064   4075   4099   4111#  4112   4123#  4129
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
                        4140    4163    4175*   4176    4189*   4193    4222    4409*   4411*   4418*   4475    4537*   4580*
                        4604*   4663    4667    4668    5117*   5118    5121*   5122    5130*   5131    5133*   5134
STN    = 000055         2244*   2647*   2671*   2686    2698*   2707    2732*   2739    2755*   2763    2805*   2814    2839*
                        2848    2873*   2882    2910*   2919    2948*   2957    2986*   2995    3026*   3072*   3118*   3123
                        3152*   3157    3186*   3192    3232*   3238    3278*   3285    3323*   3330    3368*   3375    3420*
                        3464*   3504*   3544*   3584*   3624*   3717    3725*   3729    3797*   3853*   3886*   3929*   3985*
                        3989    4053*   4057    4118*   4122    4182*   4186    4238*   4353*   4357    4405*   4438    4515*
                        4587    4594*   4608    4689*   4719    4740    4750*   4868*   4872    4917*
STPB    001150          2259*   4990*
STPFLG  001155          2259*   4990
STPS    001146          2259*   4990
STRAP   022402          2596    4994*
STRP   = 000010         4994*
STRPAD  022424          4994*
STSTM   000220          2252*
STSTNM  001102          2259*   4430    4619    4704    4982*   4984*   4985*   4986    5017    5029    5052    5273    5286
STTYIN  022232          4992*
STYPBN= ****** U        4994
STYPOS  021120          4989*   4994
STYPE   021344          4990*   4991    4994
STYPEC  021552          4990*
STYPEX  021620          4990*
STYPOC  020716          4993*   4994
STYPON  020732          4993*   4994
STYPOS  020672          4993*   4994
SUNIT   001242          2259*
SUNITM  000224          2252*
SUSWR   001254          2259*
SVECT1  001300          2259*
SVECT2  001301          2259*
SXTSTR  020122          4985*
SOFILL  021115          4988**
$40CAT= ****** U        4985    4986
.      = 034036         2247*   2251*   2252*   2259*   2596    2644    4984    4985    4986    4987*   4989*   4990    4991*
                        4992*   4993    4995    5395*   5499*   5632*   5752*   5842*   5846*   5852*
.$ASTA= ****** U        4991
.$X    = 000214         2252*
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
COMMEN  2258#
ENDCOM  2258#
ERROR   2258#  2665  2694  2719  2728  2751  2777  2789  2801  2826  2835  2860  2869  2895  2906
        2933   2944  2971  2982  3010  3022  3061  3107  3138  3148  3172  3182  3224  3270  3315
        3360   3412  3449  3456  3489  3498  3529  3538  3569  3578  3609  3618  3662  3672  3683
        3695   3706  3754  3773  3789  3817  3829  3841  3878  3920  3953  3964  4014  4034  4048
        4089   4108  4153  4172  4215  4230  4273  4296  4317  4331  4345  4387  4401  4446  4453
        4477   4498  4512  4561  4577  4636  4659  4674  4718  4739  4797  4827  4846  4862  4898
        4913   4946  4967  5277  5290
ESCAPE  2258#
MHDRS0  1622#  3986
MHDR1      3#  2648
MHDR10   350#  2874
MHDR11   405#  2911
MHDR12   448#  2949
MHDR13   491#  2987
MHDR14   534#  3027
MHDR15   633#  3073
MHDR16   732#  3119
MHDR17   773#  3153
MHDR2     45#  2672
MHDR20   814#  3187
MHDR21   859#  3233
MHDR22   904#  3279
MHDR23   946#  3324
MHDR24   988#  3369
MHDR25  1052#  3421
MHDR26  1100#  3465
MHDR27  1154#  3505
MHDR3     93#  2698
MHDR30  1199#  3545
MHDR31  1244#  3585
MHDR32  1287#  3625
MHDR33  1381#  3726
MHDR34  1432#  3798
MHDR35  1493#  3854
MHDR36  1537#  3897
MHDR37  1572#  3930
MHDR4    149#  2733
MHDR40  1675#  4054
MHDR41  1732#  4119
MHDR42  1763#  4183
MHDR43  1817#  4239
MHDR44  1883#  4354
MHDR45  1917#  4406
MHDR46  1950#  4516
MHDR47  2015#  4595
MHDR5    185#  2756
MHDR50  2047#  4690
MHDR51  2078#  4751
MHDR52  2136#  4869
MHDR53  2184#  4918
MHDR6    245#  2806
MHDR7    297#  2840
```

```
MULT    2258#
NEWTST  2258#  2647  2671  2698  2732  2755  2805  2839  2873  2910  2948  2986  3026  3072  3118
        3152   3186  3232  3278  3323  3368  3420  3464  3504  3544  3584  3624  3725  3797  3853
        3[illegible]6  3929  3985  4053  4118  4182  4238  4353  4405  4515  4594  4689  4750  4868  4917
POP     22[illegible]8#  4989  4991  4993  4995  5111
PUSH    2258#  4989  4991  4993  4995  5097
SCOPE   2258#  2647  2671  2698  2732  2755  2805  2839  2873  2910  2948  2986  3026  3072  3118
        3152   3186  3232  3278  3323  3368  3420  3464  3504  3544  3584  3624  3725  3797  3853
        3[illegible]6  3929  3985  4053  4118  4182  4238  4353  4405  4515  4594  4689  4750  4868  4917
        4972   4984
SETTRA  4994#
SETUP   2239#  2258#  2596
SKIP    2258#  2686  2690  2707  2739  2763  2814  2848  2882  2919  2957  2995  3033  3045  3079
        3091   3123  3134  3157  3168  3192  3238  3[illegible]5  3330  3375  3717  3729  3989  4057  4122
        4186   4242  4322  4357  4413  4438  4531  45[illegible]2  4587  4608  4719  4740  4764  4872  4921
SLASH   2258#
SPACE   2258#
STARS   2251   2252  2258#  2259  2647  2671  2698  2732  2755  2805  2839  2873  2910  2948  2986
        3026   3072  3118  3152  3186  3232  3278  3323  3368  3420  3464  3504  3544  3584  3624
        3725   3797  3853  3[illegible]6  3929  3985  4053  4118  4182  4238  4353  4405  4515  4594  4689
        4750   4868  4917  4984  4985  4986  4987  4988  4989  4990  4991  4992  4993  4994  4995
        4998   5000  5375  5377  5617  5619
SWRSU   2596#
TRMTRP  4994#
TYPBIN  2258#
TYPDEC  2258#  4984
TYPNUM  2258#
TYPOCS  2258#
TYPOCT  22[illegible]8#  4987
TYPTXT  2258#
$$CMRE  2259#
$$CMTM  2259#
$$ESCA  22[illegible]8#
$$NEWT  2258#  2647  2671  2698  2732  2755  2805  2839  2873  2910  2948  2986  3026  3072  3118
        3152   3186  3232  3278  3323  3368  3420  3464  3504  3544  3584  3624  3725  3797  3853
        3[illegible]6  3929  3985  4053  4118  4182  4238  4353  4405  4515  4594  4689  4750  4868  4917
$$SET   4994#
$$SKIP  2258#  2686  2707  2739  2763  2814  2848  2882  2919  2957  2995  3123  3157  3192  3238
        3285   3330  3375  3717  3729  3989  4057  4122  4186  4357  4438  4587  4608  4719  4740
        4872
.EQUAT  2239#  2258
.HEADE  22[illegible]3#  2244
.SETDH  2253#  4985
.SETUP  2238#  2595
.SWRHI  2239#  2245
.SWRLO  2239#  2245#  2246
.$ACTI  2242#  2251
.$APTB  2242#  2259#
.$APTH  2242#  2252
.$APTY  2242#  4991
.$CATC  2239#  2247
.$CMTM  2259#
.$EOP   [illegible]#  4984
.$ERRO  [illegible]#  4986
```

DZDHMA.P11 CROSS REFERENCE TABLE

M07

DZDHMA.P11 CROSS REFERENCE TABLE

DZDHMA.P11 CROSS REFERENCE TABLE

```
        4507  4523  4531  4547  4562  4578  4584  4600  4608  4629  4637  4714  4719  4740  4756
        4764  4765  4781  4798  4813  4828  4872  4886  4899  4914  4921  4934  4947  4968  49?5
        4986  4987  4988  4989  4990  4991  4992  4993  4995  5075  5082  5176  5178  5196  5204
        5219  5231  5234  5243  5250  5253  5262
CLC     3979  4524  4601  4757
CLR     2593  2594  2596  2611  2682  2711  2721  2742  2743  2768  2818  2828  2852  2862  2896
        2897  2923  2962  3000  3036  3048  3066  3082  3094  3112  3140  3174  3199  3215  3245
        3261  3297  3300  3342  3345  3381  3403  3485  3486  3495  3496  3525  3526  3535  3536
        3565  3566  3575  3576  3605  3?06  3615  3616  3649  3680  3702  3714  3738  3757  3763
        3767  3775  3783  3845  3849  3?50  3934  3993  3999  4026  4063  4068  4128  4133  4188
        4199  4256  4312  4326  4371  4409  4489  4503  4518  4521  4538  4543  4551  4570  45?8
        4598  4625  4699  4710  4754  4777  4809  4821  48?2  4893  4930  4984  49?5  4997  49?3
        4989  4993  4995  5006  5050  5083  5117  5119  5120  5130  5143  5301  5304  5324  5327
        5340  5363
CLRB    3197  3?  8  3243  3254  3288  3299  3333  3344  3377  3393  3971  4982  4985  4989  4990
        4991  4992  4993  5074  5142  5149  5161  5331
CMP     2596  2614  2658  2663  2685  2715  2773  2?85  2797  2822  2856  2891  2902  2929  2940
        2967  2978  3006  3018  3?53  3099  3133  3167  3216  3?82  3307  3352  3404  3687  3764
        3780  3792  3823  3835  3873  3?31  3915  3?23  3960  4094  4112  4158  4176  4223  4233
        4277  4289  4301  4335  4391  4412  4462  4566  4647  4663  4728  4742  4851  4957  4985
        4989  4992  5009  5122  5134  5159  5172  5174  5229  5232  5241  5248  5251  5260
CMPB    3201  3213  3227  3247  3259  3273  3292  3304  3318  3337  3349  3363  3383  3401  3415
        3761  3777  4037  4644  4778  4830  49?5  45?6  4990  4991  4992  4993
COM     2607  4586
COMB    5072
DEC     3810  3863  3867  3897  3900  3906  3908  3941  3944  4020  4411  4580  4702  4955  4984
        4987  5155  5315  5337
DECB    49 8  4990
EMT     2258
HALT    2247  4986  4990  4995  5225
INC     2625  3779  3791  3866  3890  3922  3938  3?69  4232  4299  4418  4481  4564  4701  4727
        4974  4984  4985  4986  4988  4989  4991  49?5  5?08  5121  5133  5151  5152  5313  5333
        5334
INCB    3200  3226  3246  3272  3291  3317  3336  3362  3382  3414  4480  4677  4985  4986  4990
        5076
IOT     2258
JMP     2247  2248  2249  2616  2617  4244  4347  4433  4499  4513  4590  4660  4675  4685  4686
        4766  4847  4853  4863  4983  4984  5185  5188  5226  5280  5293
JSR     2610  2637  2642  2662  2692  2718  2727  2749  2776  2788  2?00  2825  2834  2859  2868
        2894  2905  2932  2943  2970  2981  3009  3021  3032  3044  3056  3057  3078  3090  3102
        3103  3122  3126  3136  3146  3156  3160  3170  3180  3193  3219  3220  3239  3265  3?76
        3284  3310  3311  3329  3355  3356  3?76  3407  3408  3438  3442  3446  3448  3453  3?55
        3461  3476  3480  3483  3488  3492  3497  3501  3516  3520  3523  3528  3532  3537  3541
        3556  3560  3563  3568  3572  3577  3581  3?96  3600  3?33  3608  3612  3617  3621  3631
        3638  3639  3646  3651  3655  3?58  3659  3?78  3670  3671  3678  3?91  3?72  3?80  3693
        3694  3701  3704  3705  3715  3719  3728  3731  3732  3741  3745  3750  3751  3772  3788
        3803  3808  3813  3816  3826  3828  3838  3840  3844  3876  3918  3950  3951  3952  3959
        3962  3963  3998  4002  4005  4010  4011  4025  4030  4031  4040  4044  4045  4056  4071
        4085  4086  4098  4104  4105  4121  4136  4149  4150  4162  4168  4169  4185  42?2  4206
        4211  4212  4226  4227  4241  4259  4263  4269  4270  42?3  4283  4310  4313  43?4  4324
        4327  4328  4338  4341  4342  4356  4374  4378  4383  43?4  4394  4397  4398  44?0  4423
        4432  4436  4437  4465  4470  4471  4474  4492  4506  4530  4532  4546  4550  4557  4558
        4569  4573  4574  4607  4612  4628  4633  4650  4655  4656  46?6  4670  4671  4713  4731
        4735  4736  4763  4768  4780  4784  4792  4793  4812  4816  46?2  4823  4833  4841  4842
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
            4986  4988  4989  4990  4991  4992  4993  4994  5007  5017  5029  5051  5052  5100  5107
            5110  5118  5131  5150  5273  5286  5332
NEG         4988  4989
NOP         3443  3481  3521  3561  3601  3650  4349  4864  4984
RESET       4984  5278  5291
ROL         3980  4525  4602  4758  4988  4993
RTI         3451  3458  4484  4985  4986  4988  4989  4990  4992  4993  4995  5349  5357  5369
RTS         3722  3973  4987  4990  4991  4994  5011  5023  5037  5045  5058  5085  5112  5125  5137
            5163  5180  5246  5265  5305  5318  5341  5350  5358  5372
SEC         3977  4522  4599  4755
SUB         2654  3893  4410  4536  4616  4986  4989  4991
SWAB        3872  3914  3970  4076  4100  4143  4166  4221  4284  4297  4361  4458  4640  4723  4791
            4840  4951
TRAP        4994  5366
TST         2596  2603  2608  2612  2657  2681  3665  3739  3905  3947  3956  3975  4069  4307  4321
            4490  4504  4680  4810  4883  4931  4985  4986  4987  4988  4989  4990  4991  4993  4994
            5221
TSTB        3034  3046  3080  3092  3427  3471  3511  3551  3591  3945  4134  4200  4257  4372  4442
            4544  4683  4711  4901  4985  4989  4990  4991  4992  5070
.ASCII      2259
.ASCIZ      2259  4984  4987  4995  5623  5624  5630  5631  5637  5641  5645  5649  5653  5657  5661
            5665  5669  5673  5677  5681  5685  5686  5690  5694  5698  5702  5703  5707  5711  5715
            5719  5720  5726  5730  5731  5735  5736  5742  5746  5750  5751  5757  5761  5765  5769
            5773  5777  5781  5785  5789  5790  5796  5799  5803  5804  5808  5812  5816  5820  5824
            5828  5829  5830  5831  5832  5833  5834  5835  5836  5838
.BLKB        4982  5852
.BLKW       4989  5395  5499  5842  5846
.BYTE       2259  4984  4986  4988  4991  4992  5592  5593
.ENABL      2243
.END        5854
.ENDC       2244  2245  2247  2251  2252  2258  2259  2595  2596  2647  2671  2686  2690  2698  2707
            2732  2739  2755  2763  2805  2814  2839  2848  2873  2882  2910  2919  2948  2957  2986
            2995  3026  3033  3045  3072  3079  3091  3118  3123  3134  3152  3157  3168  3186  3192
            3232  3238  3278  3285  3323  3330  3368  3375  3420  3464  3504  3544  3584  3624  3717
            3725  3729  3797  3853  3886  3929  3985  3989  4053  4057  4118  4122  4182  4186  4238
            4242  4322  4353  4357  4405  4413  4438  4515  4531  4582  4587  4594  4608  4689  4719
            4740  4750  4764  4868  4872  4917  4921  4984  4985  4986  4987  4988  4989  4990  4991
            4992  4993  4994  4995  4998  5000  5375  5377  5617  5619
.EQUIV      2258
.EVEN       2259  4987  4991  4995  5625  5632  5721  5737  5752  5791  5825  5839  5851
.IF         2244  2245  2247  2251  2252  2258  2259  2595  2596  2647  2671  2686  2690  2698  2707
            2732  2739  2755  2763  2805  2814  2839  2848  2873  2882  2910  2919  2948  2957  2986
            2995  3026  3033  3045  3072  3079  3091  3118  3123  3134  3152  3157  3168  3186  3192
            3232  3238  3278  3285  3323  3330  3368  3375  3420  3464  3504  3544  3584  3624  3717
            3725  3729  3797  3853  3886  3929  3985  3989  4053  4057  4118  4122  4182  4186  4238
            4242  4322  4353  4357  4405  4413  4438  4515  4531  4582  4587  4594  4608  4689  4719
            4740  4750  4764  4868  4872  4917  4921  4984  4985  4986  4987  4988  4989  4990  4991
            4992  4993  4994  4995  4998  5000  5375  5377  5617  5619
.IFF        2245  2251  2252  2258  2259  2596  2647  2671  2686  2690  2698  2707  2732  2739  2755
            2763  2805  2814  2839  2848  2873  2882  2910  2919  2948  2957  2986  2995  3026  3033
            3045  3072  3079  3091  3118  3123  3134  3152  3157  3168  3186  3192  3232  3238  3278
            3285  3323  3330  3368  3375  3420  3464  3504  3544  3584  3624  3717  3725  3729  3797
            3853  3886  3929  3985  3989  4053  4057  4118  4122  4182  4186  4238  4242  4322  4353
            4357  4405  4413  4438  4515  4531  4582  4587  4594  4608  4689  4719  4740  4750  4764
```

DZDHMA.P11 CROSS REFERENCE TABLE

```
             4868  4872  4917  4921  4984  4985  4986  4987  4988  4989  4990  4991  4992  4993  4994
             4995  4998  5000  5375  5377  5617  5619
.IFT         4985  4986  4992  4993
.IFTF        4985  4986  4992  4993
.IIF         2244  2245  2246  2247  2259  2596  4984  4985  4986  4987  4990  4992  4993  4994
.IRP         2595  2647  2671  2698  2732  2755  2805  2839  2873  2910  2948  2986  3026  3072  3118
             3152  3186  3232  3278  3323  3368  3420  3464  3504  3544  3584  3624  3725  3797  3853
             3886  3929  3985  4053  4118  4182  4238  4353  4405  4515  4594  4689  4750  4868  4917
             4985  4989  4991  4993  4995  5097  5111
.LIST        2233  2236  2245  2247  2258  2259  2595  2596  2647  2671  2698  2732  2755  2805  2839
             2873  2910  2948  2986  3026  3072  3118  3152  3186  3232  3278  3323  3368  3420  3464
             3504  3544  3584  3624  3725  3797  3853  3886  3929  3985  4053  4118  4182  4238  4353
             4405  4515  4594  4689  4750  4868  4917  4984  4985  4986  4992  4994
.MACRO          3    45    93   149   185   245   297   350   405   448   491   534   633   732   773
              814   859   904   946   988  1052  1100  1154  1199  1244  1287  1361  1432  1493  1537
             1572  1622  1675  1732  1763  1817  1883  1917  1950  2015  2047  2078  2136  2184  2245
             2253  2259  4994
.MCALL       2238  2239  2240  2241  2242  2258  2259  2596
.MLIST          1  2235  2245  2247  2258  2259  2595  2596  2647  2671  2698  2732  2755  2805  2839
             2873  2910  2948  2986  3026  3072  3118  3152  3186  3232  3278  3323  3368  3420  3464
             3504  3544  3584  3624  3725  3797  3853  3886  3929  3985  4053  4118  4182  4238  4353
             4405  4515  4594  4689  4750  4868  4917  4984  4985  4986  4992  4994
.PAGE        2259  2592  2598  2646  2670  2697  2731  2754  2804  2838  2872  2909  2947  2985  3025
             3071  3117  3151  3185  3231  3277  3322  3367  3419  3463  3503  3543  3583  3623  3724
             3796  3852  3935  3988  3984  4052  4117  4181  4237  4352  4404  4514  4593  4688  4749
             4867  4916  4971  4997  5616
.REM         2648  2672  2699  2733  2756  2806  2840  2874  2911  2949  2987  3027  3073  3119  3153
             3187  3233  3279  3324  3369  3421  3465  3505  3545  3585  3625  3726  3798  3854  3887
             3930  3986  4054  4119  4183  4239  4354  4406  4516  4595  4690  4751  4869  4918
.REPT        2247  2259
.SBTTL       2245  2247  2251  2252  2258  2259  2647  2671  2698  2732  2755  2805  2839  2873  2910
             2948  2986  3026  3072  3118  3152  3186  3232  3278  3323  3368  3420  3464  3504  3544
             3584  3624  3725  3797  3853  3886  3929  3985  4053  4118  4182  4238  4353  4405  4515
             4594  4689  4750  4868  4917  4984  4985  4986  4987  4988  4989  4990  4991  4992  4993
             4994  4995
.TITLE       2244
.WORD        2247  2251  2252  2259  4984  4987  4988  4990  4991  4993  4995  5626  5633  5722  5738
             5753  5792
```

ERRORS DETECTED: 0

```
*DZDHMA,DZDHMA/CRF=DHMMAC,DZDHMA
RUN-TIME:  40 30  4 SECONDS
CORE USED:  37K
```

E08

Spooler runtime 35 Seconds, 216 KCS, 1371 disk reads, 1 disk writes, 230 pages printed