RP11E

DISKLESS LOGIC TEST

MD-11-DZRPW-B

EP-DZRPW-B-DL-A

COPYRIGHT © 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.

```
                                .REM    %
```

I D E N T I F I C A T I O N

PRODUCT CODE: MAINDEC-11-DZRPW-B-D

PRODUCT NAME: RP11E DISKLESS LOGIC TEST

DATE CREATED: MAY 21,1976

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: C. HESS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

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

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

COPYRIGHT (C) 1975, 1976 BY DIGITAL EQUIPMENT CORPORATION.

CONTENTS

1. ABSTRACT

THE RP11E DISKLESS TEST EXERCISES THE RP11E IN THE MAINTENANCE MODE. THE PROGRAM VERIFIES THE LOGIC CONTAINED IN THE RP11E BY UTILIZING THE THREE MAINTENANCE REGISTERS WHICH SIMULATE THE SIGNALS PASSING BETWEEN THE RP11E AND AN RP02, RPR02, OR RP03 DISK DRIVE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH 8K OF MEMORY (WITH OR WITHOUT HARDWARE SWITCH REGISTER); CONSOLE TELETYPE; RP11E DISK CONTROLLER.

2.2 STORAGE

THIS PROGRAM WILL LOAD AND RUN IN 8K.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

THE PROGRAM MAY BE LOADED FROM EITHER PAPER TAPE OR AN 'XXDP' DEVICE. REFER TO EITHER THE STANDARD PROCEDURES FOR LOADING 'ABS' FORMAT PAPER TAPES OR TO THE 'XXDP' SYSTEM REFERENCE DOCUMENT.

4. STARTING PROCEDURE

*** BEFOR STARTING REFER TO SECTION 5. ***

4.1 STARTING ADDRESSES

200 NORMAL STARTING ADDRESS
204 SELECT RP11 ADDRESS

4.1.1 START FROM LOCATION 200

WHEN THE PROGRAM IS STARTED FROM LOCATION 200, THE PROGRAM WILL PERFORM ALL OF THE TESTS IN SEQUENCE.

4.1.2 START FROM LOCATION 204

WHEN THE PROGRAM IS STARTED FROM LOCATION 204, OPERATION IS THE SAME AS THE START FROM 200, EXCEPT THAT THE PROGRAM ASKS FOR A NEW RP11E ADDRESS, VECTOR ADDRESS, AND PRIORITY. THE OPERATOR

```
                                        MAY ENTER NEW ADDRESS VALUES OR RETAIN THE OLD VALUES.  REFER TO
                                        SECTION 4.3.
```

4.2 OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3.)
2. POWER DOWN OR CYCLE DOWN DRIVE 0 (SEE SECTION 5.2).
3. LOAD ADDRESS 200 OR 204.
4. SET SWITCHES (SEE SECTION 5.1) (SOFTWARE SWITCH REGISTER IS LOC. 176)
5. PRESS START.

4.3 UNIBUS ADDRESSES

THE PROGRAM ASSUMES THE FOLLOWING UNIBUS ADDRESSES. THESE ADDRESSES MAY BE CHANGED AT THE INDICATED LOCATION BEFORE STARTING THE PROGRAM.

MEMORY LOCATION	CONTENTS	FUNCTION
1136	177560	TTY KEYBOARD STATUS REG
1140	177562	TTY KEYBOARD BUFFER REG
1142	177564	TTY PRINTER STATUS REGISTER
1144	177566	TTY PRINTER BUFFER REG

RP11E ADDRESSES (UNIBUS AND VECTOR) ARE CHANGED IN RESPONSE TO THE PROGRAM'S TYPED REQUESTS IF THE PROGRAM IS STARTED FROM LOCATION 204 OR IF THE THERE IS NO RESPONSE WHEN LOCATION 176710 IS ADDRESSED.

5. OPERATING PROCEDURES

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G (↑G); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>.
(ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
REGISTER CONTENTS WILL NOT BE CHANGED.

B) IF A CONTROL U <↑U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
BACK TO STEP 2.

5.1 OPERATIONAL SWITCH SETTINGS

WITH ALL SWITCHES SET TO ZERO, THE PROGRAM WILL TYPE ALL
ERRORS AND CONTINUE TESTING.

THE SWITCH SETTINGS ARE:

SW<15>=1...HALT ON ERROR
SW<14>=1...LOOP ON TEST
SW<13>=1...INHIBIT ERROR TYPEOUTS
SW<11>=1...INHIBIT ITERATIONS
SW<10>=1...RING BELL ON ERROR
SW<09>=1...LOOP ON ERROR
SW<08>=1...LOOP ON THE TEST NUMBER CONTAINED IN SWITCHES <0:6>
SW<07>=1...IN FLOATING 1'S & 0'S TESTS AND SILO DATA TEST, GO TO
THE NEXT TEST AFTER A DATA COMPARSION ERROR.

5.2 CONTROLLER SETUP

BEFORE THE PROGRAM IS STARTED, THE CONTROLLER MAINTENANCE SWITCH
MUST BE SET TO 'ENABLE'. DRIVE 0 MUST BE EITHER CYCLED DOWN OR
BE POWERED DOWN.

AS THE CONTROLLER USES THE 'SURP03' LINE COMING FROM THE DRIVE TO
DETERMINE THE MAXIMUM CYLINDER ADDRESS FOR THE DRIVE, SPECIAL
SETUP PROCEDURES ARE REQUIRED TO TEST THE CONTROLLER WHEN MIXED
DRIVES (RP02 OR RP03 DRIVES) ARE CONNECTED. IF DRIVE 0 IS AN RP02
OR AN RPR02 AND IT IS DESIRED TO TEST THE CONTROLLER USING THE
RP03 MAXIMUM CYLINDER ADDRESS, POWER DOWN DRIVE 0 AND CONNECT A
JUMPER BETWEEN H13P2 AND GROUND. GROUNDING THIS PIN FORCES THE
RP03 SIGNAL ('SURP03') HIGH.

IF DRIVE IS AN RP03 AND THE CONTROLLER IS TO BE TESTED USING THE RP02
MAXIMUM CYLINDER ADDRESS, POWER DOWN DRIVE 0.

IF THE CONTROLLER IS TO BE TESTED USING THE CYLINDER ADDRESS LIMIT
OF DRIVE 0, THE DRIVE MUST BE CYCLED DOWN.

5.3 TEST SELECTION

TO EXECUTE A SPECIFIC TEST, SET SW<08> TO 1 AND SET THE TEST NUMBER
(REFER TO THE PROGRAM LISTING) IN SWITCHES <0:6>, START THE PROGRAM
IN THE USUAL MANNER. THE PROGRAM WILL PERFORM ALL THE TESTS UP TO
THE SPECIFIED TEST FOR ONE ITERATION AND WILL LOOP ON THE TEST SPECIFIED
IN THE SWITCHES. (REFERE TO SECTION 5. FOR SOFTWARE SWITCH REGISTER)

5.4 CHANGING RP11E UNIBUS AND VECTOR ADDRESSES

THE OPERATOR MAY CHANGE THE DEFAULT VALUES OF THE RP11E UNIBUS AND VECTOR ADDRESSES BY STARTING THE PROGRAM FROM LOCATION 204. THE PROGRAM WILL TYPE THE FOLLOWING MESSAGE:

```
RPADR = 176710/
```

THE OPERATOR MAY EITHER ENTER A NEW RP11E UNIBUS ADDRESS VALUE OR HE MAY RETAIN THE PRELOADED VALUE BY ENTERING A CARRIAGE RETURN.

THE PROGRAM WILL THEN TYPE:

```
RPVEC = 254/
```

THE VECTOR ADDRESS MAY BE CHANGED BY ENTERING A NEW VALUE OR THE PRELOADED VALUE MAY BE RETAINED BY ENTERING A CARRIAGE RETURN.

THE PROGRAM WILL THEN REQUEST A NEW RP11E PRIORITY WITH THE FOLLOWING MESSAGE:

```
RPPRIO = 5/
```

A NEW PRIORITY MAY BE ENTERED OR THE PRESENT PRIORITY MAY BE RETAINED.

THE PROGRAM WILL THEN VERIFY THAT THE LOADED RP11E UNIBUS ADDRESS RESPONDS WHEN THE PROGRAM ADDRESSES THE LOCATION. IF THERE IS NO RESPONSE FROM THE ADDRESS, THE PROGRAM WILL TYPE:

```
'RP11E DIDN'T RESPOND TO ADDRESSING'
```

THE PROGRAM WILL LOOP BACK TO THE RP11E ADDRESS ENTRY MESSAGE.

THE PROGRAM ALWAYS CHECKS FOR RESPONSE FROM THE RP11E BEFORE STARTING THE TESTS WHETHER OR NOT A START FROM LOCATION 204 WAS INITIATED.

6. ERRORS

WHEN THE PROGRAM DETECTS AN ERROR, THE ERROR ROUTINE IS CALLED AND THE SW(13) IS NOT SET, THE ERROR MESSAGE PERTAINING TO THE ERROR WILL BE TYPED. EACH ERROR TYPEOUT WILL CONTAIN THE FOLLOWING*

A. AN ERROR MESSAGE
B. A DATA HEADER LINE
C. A DATA LINE CONTAINING:

   1. THE TEST NUMBER
   2. THE PC (PROGRAM COUNTER CONTENTS) WHERE THE ERROR CALL WAS MADE.
   3. CONTENTS OF THE APPROPRIATE REGISTERS

```
                                        D.  THE ERROR ROUTINE WILL HONOR A ^G (REFERENCE OPERATING PROCEDURE
                                            SECTION 5.)
```

7. MISCELLANEOUS

7.1 EXECUTION TIME

PASS 1 OF THE PROGRAM TAKES ABOUT 30 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE ABOUT 3 MINUTES.

7.2 END OF TEST

ONE TEST ITERATION IS DEFINED AS 8 PASSES OF THE PROGRAM. AT END OF TEST, THE PROGRAM WILL TYPEOUT 'END OF TEST' AND CONTINUE OPERATION.

7.3 STACK POINTER

THE STACK POINTER IS INITIALLY SET TO 1100 AND EXTENDS DOWNWARD IN MEMORY.

7.4 SUBROUTINE CALLS

THE SUBROUTINE CALLS USED BY THE PROGRAM ARE:

A. 'SCOPE' (IOT) INSTRUCTION). THIS CALL IS PLACED BETWEEN EACH TEST IN THE PROGRAM. THIS POUTINE ESTABLISHES THE TEST ITERATION COUNT AND LOOP ON TEST AND LOOP ON ERROR ADDRESSES. ↑G IS HONORED. (REFER TO SECTION 5. FOR OPERATOR OPTIONS)

B. 'ERROR' (EMTINSTRUCTION. THIS CALL IS USED TO REPORT ALL ERRORS. A CALL TO THIS ROUTINE IS FOLLOWED BY A NUMBER WHICH IDENTIFIES THE ERROR MESSAGE WHICH WILL BE TYPED. ↑G IS HONORED. (REFER TO SECTION 5. FOR OPERATOR OPTIONS)

THE TRAP INSTRUCTION IS USED FOR THE FOLLOWING SUBROUTINE CALLS:

TYPE - TTY TYPEOUT ROUTINE
TYPOC - TYPE OCTAL NUMBER (WITH LEADING ZEROS)
TYPOS - TYPE OCTAL NUMBER (NO LEADING ZEROS)
RDCHR - READ A CHARACTER FROM THE TTY KEYBOARD
RDLIN - READ A LINE FROM THE TTY KEYBOARD
RDOCT - READ AN OCTAL NUMBER FROM THE TTY KEYBOARD
SAVREG - ROUTINE TO SAVE R0 - R5
RESREG - ROUTINE TO RESTORE R0 - R5

8. TEST DESCRIPTION

8.1 BEFORE TESTING IS STARTED, THE PROGRAM VERIFIES THAT THE ADDRESS SPECIFIED AS THE FIRST RP11E ADDRESS RESPONDS WHEN ADDRESSED. THE PROGRAM WILL NOT CONTINUE UNTIL IT RECEIVES A RESPONSE FROM THE RP11E.

8.2 TEST 0 - THIS TEST VERIFIES THAT ALL LOADABLE BITS IN REGISTERS RPER, RPCS, RPWC, RPBA, RPCA, AND RPDA CAN BE SET AND THAT A 'RESET' INSTRUCTION FROM THE PROCESSOR CLEARS THE REGISTERS.

8.3 TEST 1 - THIS TEST VERIFIES THAT THE SELECTED UNIT ON LINE ('SUOL') BIT IN RPDS CAN BE SET IN MAINTENANCE MODE.

8.4 TEST 2 - THIS TEST VERIFIES THAT THE RP11E INITIALIZE INSTRUCTION CLEARS THE REGISTERS. THE REGISTERS CHECKED ARE RPER, RPCS, RPWC, RPBA, RPCA, AND RPDA.

8.5 TEST 3 - THIS TEST TESTS THE LOADING OF ALL POSSIBLE BITS IN REGISTER RPER USING A FLOATING 1'S AND 0'S PATTERN.

8.6 TEST 4 - THIS TEST TESTS THE LOADING OF ALL POSSIBLE BITS IN REGISTER RPCS USING A FLOATING 1'S AND 0'S PATTERN.

8.7 TEST 5 - THIS TEST TESTS THE LOADING OF ALL POSSIBLE BITS IN REGISTER RPWC USING A FLOATING 1'S AND 0'S PATTERN.

8.8 TEST 6 - THIS TEST TESTS THE LOADING OF ALL POSSIBLE BITS IN REGISTER RPBA USING A FLOATING 1'S AND 0'S PATTERN.

8.9 TEST 7 - THIS TEST TESTS THE LOADING OF ALL POSSIBLE BITS IN REGISTER RPCA USING A FLOATING 1'S AND 0'S PATTERN.

8.10 TEST 10 - THIS TEST TESTS THE LOADING OF ALL POSSIBLE BITS IN REGISTER RPDA USING A FLOATING 1'S AND 0'S PATTERN.

8.11 TEST 11 - THIS TEST VERIFIES THE HIGH SPEED OPERATION OF THE WORD COUNT REGISTER IN A MANNER SIMILAR TO ITS OPERATION AS A COUNTER DURING A DATA TRANSFER OPERATION. THE CONTENTS OF EACH MEMORY LOCATION (UP TO 28K) ARE LOADED INTO AND READ FROM THE REGISTER AND CHECKED. THE LOAD AND READ OF THE REGISTER ARE PERFORMED AT THE MAXIMUM POSSIBLE PROGRAM RATE. FAILURES IN THIS TEST CAN INDICATE SETTLE DOWN PROBLEMS IN THE REGISTER

8.12 TEST 12 - THIS TEST VERIFIES THE HIGH SPEED OPERATION OF THE BUFFER ADDRESS REGISTER IN A MANNER SIMILAR TO ITS OPERATION AS A COUNTER DURING A DATA TRANSFER OPERATION. THE CONTENTS OF EACH MEMORY LOCATION (UP TO 28K) ARE LOADED INTO AND READ FROM THE REGISTER AND CHECKED. THE LOAD AND READ OF THE REGISTER ARE PERFORMED AT THE MAXIMUM POSSIBLE PROGRAM RATE. FAILURES IN THIS TEST CAN INDICATE SETTLE DOWN PROBLEMS IN THE REGISTER

8.13 TEST 13 - TEST THAT THE 'SURDY' BIT CAN BE SET AND CLEARED. THE BIT IS TESTED BY LOADING THE DRIVE READY BIT IN THE MAINTENANCE REGISTER

```
418
419                             8.14    TEST 14 - TEST THAT 'SELECTED UNIT SEEK INCOMPLETE' CAN BE SET IN
420                                     MAINTENANCE MODE, THAT 'DRIVE ERROR' SETS IN RPER, AND THAT
421                                     'ERR' AND 'ME' SET IN RPCS.
422
```

8.15 TEST 15 - TEST THAT 'SELECTED UNIT FILE UNSAFE' CAN BE SET AND CLEARED IN MAINTENANCE MODE.

8.16 TEST 16 - TEST THAT 'SELECTED UNIT WRITE PROTECT' CAN BE SET AND CLEARED IN MAINTENANCE MODE.

8.17 TEST 17 - TEST THAT 'SELECTED UNIT WRITE PROTECTED' CAN BE SET AND CLEARED IN MAINTENANCE MODE.

8.18 TEST 20 - TEST THAT EACH OF THE ATTENTION BITS CAN BE SET AND EACH IS CLEARED WHEN A '1' IS WRITTEN INTO THE BIT. VERIFY THAT A 'RESET' INSTRUCTION CLEARS THE ATTENTION BITS AND THAT WRITING 0'S INTO THE ATTENTION BITS DOES NOT CLEAR THE BITS.

8.19 TEST 21 - VERIFY THAT 'WPV' OCCURS WHEN A WRITE COMMAND IS ISSUED TO THE CONTROLLER AND 'SUWP' IS SET. VERIFY THAT 'ERR' AND 'HE' SET IN RPCS.

8.20 TEST 22 - VERIFY THAT 'FUV' OCCURS WHEN A COMMAND IS ISSUED TO THE CONTROLLER AND 'SUFU' IS SET. VERIFY THAT 'ERR' AND 'HE' BITS SET IN RPCS.

8.21 TEST 23 - VERIFY THAT 'NXC' DOES NOT SET WHEN A WRITE COMMAND IS ISSUED USING EACH VALID CYLINDER CODE.

8.22 TEST 24 - VERIFY THAT 'NXC' SETS WHEN A WRITE ORDER IS GIVEN FOR EACH INVALID CYLINDER ADDRESS. VERIFY THAT 'ERR' AND 'HE' BITS SET IN RPCS

8.23 TEST 25 - VERIFY THAT 'NXT' DOES NOT SET WHEN A WRITE COMMAND IS GIVEN USING EACH VALID TRACK ADDRESS.

8.24 TEST 26 - VERIFY THAT 'NXT' OCCURS WHEN A WRITE COMMAND IS GIVEN USING EACH INVALID TRACK ADDRESS CODE UP TO THE LIMIT OF THE REGISTER. VERIFY THAT 'ERR' AND 'HE' BITS SET IN RPCS.

8.25 TEST 27 - VERIFY THAT 'NXS' DOES NOT OCCUR WHEN A WRITE COMMAND IS GIVEN USING EACH VALID SECTOR ADDRESS.

8.26 TEST 30 - VERIFY THAT 'NXS' OCCURS WHEN A WRITE COMMAND IS GIVEN USING EACH INVALID SECTOR ADDRESS CODE UP TO THE LIMIT OF THE REGISTER. VERIFY THAT 'ERR' AND 'HE' BITS SET IN RPCS.

8.27 TEST 31 - VERIFY THAT A 'PROG' ERROR OCCURS IF A COMMAND IS ISSUED WHEN 'SUOL' IS NOT SET. VERIFY THAT 'ERR' AND 'HE' BITS ARE SET IN RPCS.

8.2[illegible] TEST 32 - VERIFY THAT A 'MODE' ERROR OCCURS IF AN ATTEMPT IS MADE TO WRITE A HEADER WHEN THE CONTROLLER IS IN 'PDP-11' MODE.

8.29 TEST 33 - VERIFY THAT AN INTERRUPT OCCURS WHEN AN ATTENTION BIT AND 'AIE' ARE SET. WHEN THE INTERRUPT OCCURS, VERIFY THAT 'AIE' HAS RESET.

8.30 TEST 34 - VERIFY THAT THE CONTROLLER DOES NOT GENERATE AN INTERRUPT WHEN 'AIE' IS SET AND NO ATTENTION BITS ARE SET.

```
   479
```

8.31 TEST 35 - VERIFY THAT EACH ATTENTION BIT SET WILL GENERATE A SEPARATE INTERRUPT. ATTENTION BITS 0 AND 1 ARE SET AND 'AIE' IS SET; VERIFY THAT AN INTERRUPT OCCURS. RESET ATTENTION BIT 0 AND SET 'AIE' AGAIN; VERIFY THAT A SECOND INTERRUPT OCCURS.

8.32 TEST 36 - VERIFY THAT AN INTERRUPT OCCURS WHEN THE CONTROLLER IS READY AND 'IDE' IS SET. VERIFY THAT 'IDE' DOES NOT RESET WHEN AN INTERRUPT OCCURS.

8.33 TEST 37 - VERIFY THAT AN INTERRUPT DOES NOT OCCUR WHEN THE CONTROLLER IS READY AND 'IDE' IS NOT SET.

8.34 TEST 40 - VERIFY THAT A CONTROLLER INTERRUPT OCCURS WITH THE PROCESSOR AT PRIORITY 4.

8.35 TEST 41 - VERIFY THAT A CONTROLLER INTERRUPT DOES NOT OCCUR WHEN THE PROCESSOR IS AT PRIORITY 5.

8.36 TEST 42 - VERIFY THAT A CONTROLLER INTERRUPT DOES NOT OCCUR WHEN THE PROCESSOR IS AT PRIORITY 6.

8.37 TEST 43 - VERIFY THAT A CONTROLLER INTERRUPT DOES NOT OCCUR WHEN THE PROCESSOR IS AT PRIORITY 7.

8.38 TEST 44 - VERIFY THAT CONTROLLER READY RESETS WHEN 'GO' IS SET AND THAT CONTROLLER INITIALIZE RETURNS THE CONTROLLER TO THE 'READY' STATE.

8.39 TEST 45 - TEST THE 'SUCA' BIT PATH IN THE CONTROLLER.

8.40 TEST 46 - VERIFY THAT THE 'SOT' COUNTER COUNTS SECTOR PULSES PROPERLY.

8.41 TEST 47 - VERIFY THAT 'INDEX' CLEARS THE 'SOT' COUNTER. CHECK THAT THE 'SOT' CONTAINS SOME COUNT; IF THE 'SOT' IS CLEAR, ISSUE A SECTOR PULSE TO SET A VALUE INTO THE COUNTER. GENERATE AN INDEX PULSE AND VERIFY THAT THE 'SOT' CLEARS.

8.42 TEST 50 - VERIFY THAT SILO MEMORY LOGIC. CLEAR THE CONTROLLER; VERIFY THAT THE SILO'S 'INPUT READY' BIT IS SET. LOAD THE SILO AND VERIFY THAT WORD 'BUBBLES' THROUGH THE SILO AND THAT 'OUTPUT READY' SETS. READ THE SILO REGISTER AND VERIFY THAT THE DATA IS CORRECT. AFTER THE SILO REGISTER IS READ, VERIFY THAT 'OUTPUT READY' CLEARS. PERFORM THE TEST USING FLOATING 1'S AND 0'S PATTERN.

8.43 TEST 51 - VERIFY THAT THE SILO MEMORY CAN HOLD 64 WORDS. LOAD THE SILO WITH NUMBERS FROM 1 (8) TO 100 (8). VERIFY THAT 'INPUT READY' CLEARS AFTER 64 (10) WORDS HAVE BEEN LOADED INTO THE SILO AND THAT 'OUTPUT READY' IS SET. READ EACH WORD FROM THE SILO AND COMPARE IT. VERIFY THAT 'INPUT READY' SET WHEN THE FIRST WORD IS READ FROM THE FULL SILO. VERIFY THAT 'INPUT READY' REMAINS SET AND THAT 'OUTPUT READY' STAYS SET UNTIL THE SILO IS EMPTY.

8.44 TEST 52 - ISSUE A SEEK IN MAINTENANCE MODE. VERIFY THAT 'SET CYLINDER', 'RESET HEAD', 'SET HEAD', AND 'SEEK START' CONTROL SIGNALS OCCUR AT THE PROPER TIME. VERIFY THAT THE HEAD ADDRESS AND CYLINDER ADDRESS ARE CORRECT AT THE PROPER TIME.

```
   536  -
```

```
                        8.45    TEST 53 - ISSUE A HOME SEEK IN MAINTENANCE MODE.  VERIFY THAT
                                'CONTROL' AND 'RESTORE' ARE SET ON THE BUS.

                        8.46    TEST 54 - ISSUE A READ IN MAINTENANCE MODE.  VERIFY THAT 'CONTROL',
                                'SELECT HEAD' AND 'READ' ARE ON THE DRIVE BUS OUT LINES.

                        8.47    TEST 55 - ISSUE A WRITE FORMAT IN MAINTENANCE MODE.  VERIFY THAT
                                'CONROL', 'SELECT HEAD', 'ERASE', AND 'WRITE' ARE SET ON THE
                                BUS OUT LINES.



                        9.      PROGRAM LISTING
                                ---------------
```

```
551
552                                             %
```

```
                                .TITLE  MAINDEC-11-DZRPW-B, RP11-E DISKLESS LOGIC TEST
                                ;*COPYRIGHT (C) 1975,1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY C. HESS/F. ROEMER
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.
                                ;*
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          ------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<6:0>
                                ;*         7            EXIT FROM TEST AFTER ERROR DURING FLOATING 1'S AND 0'
                                ;*                      TESTS AND THE SILO DATA TEST.

                                .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  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
  000200                                $SVPC=.                 ;SAVE PC
  000046                                .=46
000046  016364                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
  000052                                .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
  000200                                .=$SVPC                 ;; RESTORE PC

                                .SBTTL  STARTING ADDRESSES

  000200                        .=200
                                ;*200 = NORMAL START
000200  000137  002364                  JMP     START
                                ;*204 = SELECT RP11E ADDRESS
000204  000137  002372                  JMP     START1

                                .SBTTL  BASIC DEFINITIONS
```

```
                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
```

```
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
000002                          BIT01=  2
000001                          BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                          TBITVEC=14              ;;"T" BIT
000014                          TRTVEC= 14              ;;TRACE TRAP
000014                          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                          PWRVEC= 24              ;;POWER FAIL
000030                          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                          TRAPVEC=34              ;;"TRAP" TRAP
000060                          TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                          TPVEC=  64              ;;TTY PRINTER VECTOR
000240                          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                                .SBTTL  RP11E REGISTER INDEX EQUATES

000000                          RPOS=00
```

```
                        000002          RPER=02
                        000004          RPCS=04
                        000006          RPWC=06
                        000010          RPBA=10
                        000012          RPCA=12
                        000014          RPDA=14
                        000016          RPM1=16
                        000020          RPM2=20
                        000022          RPM3=22
                        000024          SUCA=24
                        000026          SILO=26
```

```
                                .SBTTL  COMMON TAGS

                                ;;**********************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001100                  .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG  ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001162  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001164  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001166  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001172     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001173     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001174  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;**********************************************************************
001176  000000          MAXCYL: .WORD   0               ;MAXIMUM CYLINDER ADDRESS CHECKED
001200  000000          MAXPAT: .WORD   0               ;MAXIMUM CYLINDER ADDRESS BITS CHECKED
001202  000000          MASK:   .WORD   0               ;STORAGE FOR PATTERN IN FLOATING 1'S & 0'S TESTS
001204  000000          BUSADR: .WORD   0               ;CHANGE RP11 ADDRESS FLAG
001206  000000          TPL:    .WORD   0               ;STORE BITS TO BE LOADED INTO RPM3 HERE

                        ;RP11E ADDRESSES

001210  176710          RPADR:  .WORD   176710          ;RP11E BUS ADDRESS
001212  000254  000256  RPVEC:  .WORD   254,256         ;RP11E VECTOR ADDRESSES
001216  000240          RPPRIO: .WORD   <5*32.>          ;RP11E PRIORITY
```

```
790
791                                     ;SAVE THE RP11E REGISTERS HERE
792
793  001220  000000    $RPDS:  .WORD   0       ;DRIVE STATUS REGISTER
794  001222  000000    $RPER:  .WORD   0       ;ERROR REGISTER
795  001224  000000    $RPCS:  .WORD   0       ;COMMAND & STATUS REGISTER
796  001226  000000    $RPWC:  .WORD   0       ;WORD COUNT REGISTER
797  001230  000000    $RPBA:  .WORD   0       ;BUFFER ADDRESS REGISTER
798  001232  000000    $RPCA:  .WORD   0       ;CURRENT CYLINDER ADDRESS REGISTER
799  001234  000000    $RPDA:  .WORD   0       ;TRACK-SECTOR ADDRESS REGISTER
800  001236  000000    $RPMI:  .WORD   0       ;MAINTENANCE REGISTER #1
801  001240  000000    $SUCA:  .WORD   0       ;SELECTED UNIT CYLINDER ADDRESS REGISTER
802  001242  000000    $SILO:  .WORD   0       ;SILO REGISTER
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001244                          $ERRTB:

                                ;ERROR 1

001244  022510                          EM1             ;RP11 DIDN'T RESPOND TO ADDRESSING
001246  027751                          DH1
001250  031352                          DT1
001252  031652                          DF

                                ;ERROR 2

001254  022552                          EM2             ;CAN'T LOAD REGISTER CORRECTLY
001256  027767                          DH2
001260  031356                          DT2
001262  031662                          DF2

                                ;ERROR 3

001264  022610                          EM3             ;RESET DIDN'T CLEAR REGISTER
001266  027767                          DH2
001270  031356                          DT2
001272  031662                          DF2

                                ;ERROR 4

001274  022644                          EM4             ;CONTROLLER CLEAR DIDN'T CLEAR REGISTER
001276  027767                          DH2
001300  031356                          DT2
001302  031662                          DF2

                                ;ERROR 5

001304  022713                          EM5             ;FLOATING 1'S & 0'S TEST ERROR
001306  027767                          DH2
001310  031356                          DT2
001312  031662                          DF2

                                ;ERROR 6

001314  022751                          EM6             ;REGISTER RAPID ACCESS TEST ERROR
001316  027767                          DH2
001320  031356                          DT2
```

```
001322  031662                          DF2
                                
                                ;ERROR 7
                                
001324  023012                          EM7             ;CAN'T SET 'SURDY' BIT
001326  030035                          DH7
001330  031370                          DT7
001332  031656                          DF1
                                
                                ;ERROR 10
                                
001334  023040                          EM10            ;CAN'T CLEAR THE 'SURDY' BIT
001336  030035                          DH7
001340  031370                          DT7
001342  031656                          DF1
                                
                                ;ERROR 11
                                
001344  023074                          EM11            ;CAN'T SET 'SUSI'
001346  030062                          DH11
001350  031376                          DT11
001352  031662                          DF2
                                
                                ;ERROR 12
                                
001354  023115                          EM12            ;'DSK ERR' NOT SET WITH 'SUSI'
001356  030062                          DH11
001360  031376                          DT11
001362  031662                          DF2
                                
                                ;ERROR 13
                                
001364  023153                          EM13            ;'ERR' OR 'HE' NOT SET WITH 'SUSI'
001366  030062                          DH11
001370  031376                          DT11
001372  031662                          DF2
                                
                                ;ERROR 14
                                
001374  023215                          EM14            ;CAN'T SET 'SUFU'
001376  030062                          DH11
001400  031376                          DT11
001402  031662                          DF2
                                
                                ;ERROR 15
                                
001404  023236                          EM15            ;CAN'T CLEAR 'SUFU'
001406  030062                          DH11
001410  031376                          DT11
001412  031662                          DF2
                                
                                ;ERROR 16
                                
001414  023261                          EM16            ;CAN'T SET 'SUWP'
001416  030062                          DH11
001420  031376                          DT11
```

```
915  001422  031662                          DF2
916
917                                  ;ERROR 17
918
919  001424  023302                          EM17            ;CAN'T CLEAR 'SUMP'
920  001426  030062                          DH11
921  001430  031376                          DT11
922  001432  031662                          DF2
923
924                                  ;ERROR 20
925
926  001434  023325                          EM20            ;CAN'T SET ATTENTION BIT
927  001436  030127                          DH20
928  001440  031410                          DT20
929  001442  031666                          DF20
930
931                                  ;ERROR 21
932
933  001444  023355                          EM21            ;CAN'T CLEAR ATTENTION BIT
934  001446  030127                          DH20
935  001450  031410                          DT20
936  001452  031666                          DF20
937
938                                  ;ERROR 22
939
940  001454  023407                          EM22            ;RESET DIDN'T CLEAR ATTENTION BITS
941  001456  030035                          DH7
942  001460  031370                          DT7
943  001462  031656                          DF1
944
945                                  ;ERROR 23
946
947  001464  023451                          EM23            ;ATTENTION BITS CLEARED BY WRITING 0'S INTO RPDS
948  001466  030035                          DH7
949  001470  031370                          DT7
950  001472  031656                          DF1
951
952                                  ;ERROR 24
953
954  001474  023531                          EM24            ;CAN'T SET 'WPV' ERROR
955  001476  030164                          DH24
956  001500  031420                          DT24
957  001502  031666                          DF20
958
959                                  ;ERROR 25
960
961  001504  023557                          EM25            ;'ERR' OR 'HE' NOT SET WITH 'WPV'
962  001506  030164                          DH24
963  001510  031420                          DT24
964  001512  031666                          DF20
965
966                                  ;ERROR 26
967
968  001514  023620                          EM26            ;CAN'T SET 'FUV' ERROR
969  001516  030164                          DH24
970  001520  031420                          DT24
```

```
  971  001522  031666                          DF20
  972
  973                                ;ERROR 27
  974
  975  001524  023646                          EM27          ;'ERR' OR 'HE' NOT SET WITH 'FUV'
  976  001526  030164                          DH24
  977  001530  031420                          DT24
  978  001532  031666                          DF20
  979
  980                                ;ERROR 30
  981
  982  001534  023707                          EM30          ;'NXC' ERROR SET WITH VALID CYLINDER ADDRESS
  983  001536  030221                          DH30
  984  001540  031430                          DT30
  985  001542  031672                          DF30
  986
  987                                ;ERROR 31
  988
  989  001544  023763                          EM31          ;'NXC' DIDN(T SET WITH INVALID CYLINDER ADDRESS
  990  001546  030221                          DH30
  991  001550  031430                          DT30
  992  001552  031672                          DF30
  993
  994                                ;ERROR 32
  995
  996  001554  024042                          EM32          ;'ERR' OR 'HE' DIDN'T SET WITH 'NXC'
  997  001556  030221                          DH30
  998  001560  031430                          DT30
  999  001562  031672                          DF30
 1000
 1001                                ;ERROR 33
 1002
 1003  001564  024106                          EM33          ;'NXT' ERROR SET WITH VALID TRACK ADDRESS
 1004  001566  030312                          DH33
 1005  001570  031444                          DT33
 1006  001572  031672                          DF30
 1007
 1008                                ;ERROR 34
 1009
 1010  001574  024157                          EM34          ;'NXT' DIDN'T SET WITH INVALID TRACK ADDRESS
 1011  001576  030312                          DH33
 1012  001600  031444                          DT33
 1013  001602  031672                          DF30
 1014
 1015                                ;ERROR 35
 1016
 1017  001604  024233                          EM35          ;'ERR' OR 'HE' DIDN'T SET WITH 'NXT'
 1018  001606  030312                          DH33
 1019  001610  031444                          DT33
 1020  001612  031672                          DF30
 1021
 1022                                ;ERROR 36
 1023
 1024  001614  024277                          EM36          ;'NXS' ERROR SET WITH VALID SECTOR ADDRESS
 1025  001616  030312                          DH33
 1026  001620  031444                          DT33
```

```
1027  001622  031672                        DF30
1028
1029                                ;ERROR 37
1030
1031  001624  024351                        EM37            ;'NXS' DIDN'T SET WITH INVALID SECTOR ADDRESS
1032  001626  030312                        DH33
1033  001630  031444                        DT33
1034  001632  031672                        DF30
1035
1036                                ;ERROR 40
1037
1038  001634  024426                        EM40            ;'ERR' OR 'HE' DIDN'T SET WITH 'NXS'
1039  001636  030312                        DH33
1040  001640  031444                        DT33
1041  001642  031672                        DF30
1042
1043                                ;ERROR 41
1044
1045  001644  024472                        EM41            ;CAN'T SET 'PROG' ERROR
1046  001646  030164                        DH24
1047  001650  031420                        DT24
1048  001652  031666                        DF20
1049
1050                                ;ERROR 42
1051
1052  001654  024521                        EM42            ;'ERR' OR 'HE' NOT SET WITH 'PROG'
1053  001656  030164                        DH24
1054  001660  031420                        DT24
1055  001662  031666                        DF20
1056
1057                                ;ERROR  43
1058
1059  001664  024563                        EM43            ;CAN'T SET 'MODE' ERROR
1060  001666  030164                        DH24
1061  001670  031420                        DT24
1062  001672  031666                        DF20
1063
1064                                ;ERROR 44
1065
1066  001674  024612                        EM44            ;'ERR' OR 'HE' NOT SET WITH 'MODE'
1067  001676  030164                        DH24
1068  001700  031420                        DT24
1069  001702  031666                        DF20
1070
1071                                ;ERROR 45
1072
1073  001704  024654                        EM45            ;CAN'T CLEAR 'SUSI'
1074  001706  030062                        DH11
1075  001710  031376                        DT11
1076  001712  031662                        DF2
1077
1078                                ;ERROR 46
1079
1080  001714  024677                        EM46            ;NO ATTENTION INTERRUPT
1081  001716  030403                        DH46
1082  001720  031460                        DT46
```

```
1083  001722  031662                          DF2
1084
1085                                  ;ERROR 47
1086
1087  001724  024726                          EM47            ;'AIE' DIDN'T CLEAR WHEN INTERRUPT OCCURED
1088  001726  030403                          DH46
1089  001730  031460                          DT46
1090  001732  031662                          DF2
1091
1092                                  ;ERROR 50
1093
1094  001734  025000                          EM50            ;NO INTERRUPT WITH ATTENTION BITS 0 & 1 SET
1095  001736  030463                          DH50
1096  001740  031472                          DT50
1097  001742  031666                          DF20
1098
1099                                  ;ERROR 51
1100
1101  001744  025053                          EM51            ;SECOND INTERRUPT DIDN'T OCCUR WITH ATTENTION BIT 1 SET
1102  001746  030463                          DH50
1103  001750  031472                          DT50
1104  001752  031666                          DF20
1105
1106                                  ;ERROR 52
1107
1108  001754  025142                          EM52            ;NO INTERRUPT FROM CONTROLLER 'READY'
1109  001756  030520                          DH52
1110  001760  031502                          DT52
1111  001762  031656                          DF1
1112
1113                                  ;ERROR 53
1114
1115  001764  025207                          EM53            ;'READY' INTERRUPT WITH 'IDE' SET
1116  001766  030520                          DH52
1117  001770  031502                          DT52
1118  001772  031656                          DF1
1119
1120                                  ;ERROR 54
1121
1122  001774  025250                          EM54            ;NO INTERRUPT FROM RP11 WITH CPU AT PR4
1123  001776  030520                          DH52
1124  002000  031502                          DT52
1125  002002  031656                          DF1
1126
1127                                  ;ERROR 55
1128
1129  002004  025317                          EM55            ;INTERRUPT FROM RP11 WITH CPU AT PR5
1130  002006  030520                          DH52
1131  002010  031502                          DT52
1132  002012  031656                          DF1
1133
1134                                  ;ERROR 56
1135
1136  002014  025363                          EM56            ;INTERRUPT FROM RP11 WITH CPU AT PR6
1137  002016  030520                          DH52
1138  002020  031502                          DT52
```

```
1139  002022  031656                        DF1
1140
1141                                ;ERROR 57
1142
1143  002024  025427                        EM57            ;INTERRUPT FROM RP11 WITH CPU AT PR7
1144  002026  030520                        DH52
1145  002030  031502                        DT52
1146  002032  031656                        DF1
1147
1148                                ;ERROR 60
1149
1150  002034  025473                        EM60            ;'READY' DIDN'T CLEAR AT END OF OPERATION
1151  002036  030520                        DH52
1152  002040  031502                        DT52
1153  002042  031656                        DF1
1154
1155                                ;ERROR 61
1156
1157  002044  025544                        EM61            ;'SUCA' INCORRECT
1158  002046  030545                        DH61
1159  002050  031510                        DT61
1160  002052  031666                        DF20
1161
1162                                ;ERROR 62
1163
1164  002054  025565                        EM62            ;INDEX DIDN'T CLEAR 'SOT'
1165  002056  030613                        DH62
1166  002060  031520                        DT62
1167  002062  031656                        DF1
1168
1169                                ;ERROR 63
1170
1171  002064  025616                        EM63            ;'SOT' DIDN'T COUNT CORRECTLY
1172  002066  030640                        DH63
1173  002070  031526                        DT63
1174  002072  031666                        DF20
1175
1176                                ;ERROR 64
1177
1178  002074  025653                        EM64            ;CONTROLLER CLEAR DIDN'T SET SILO INPUT READY
1179  002076  030705                        DH64
1180  002100  031536                        DT64
1181  002102  031666                        DF20
1182
1183                                ;ERROR 65
1184
1185  002104  025730                        EM65            ;SILO OUTPUT READY NOT SET AFTER WORD LOADED INTO SILO
1186  002106  030751                        DH65
1187  002110  031546                        DT65
1188  002112  031656                        DF1
1189
1190                                ;ERROR 66
1191
1192  002114  026016                        EM66            ;DATA READ FROM SILO IS INCORRECT
1193  002116  030776                        DH66
1194  002120  031554                        DT66
```

```
1195  002122  031672                        DF30
1196
1197                                ;ERROR 67
1198
1199  002124  026057                        EM67        ;SILO INPUT READY DIDN'T CLEAR AFTER 64 WORDS WERE
1200                                                      ;LOADED INTO THE SILO
1201  002126  030751                        DH65
1202  002130  031546                        DT65
1203  002132  031656                        DF1
1204
1205                                ;ERROR 70
1206
1207  002134  026167                        EM70        ;SILO OUTPUT READY DIDN'T CLEAR AFTER SILO EMPTIED
1208  002136  030751                        DH65
1209  002140  031546                        DT65
1210  002142  031656                        DF1
1211
1212                                ;ERROR 71
1213
1214  002144  026251                        EM71        ;BUS OUT LINES TO THE DRIVE ARE NOT CLEARED BY CONTROLLER CLEAR
1215  002146  031054                        DH71
1216  002150  031570                        DT71
1217  002152  031676                        DF71
1218
1219                                ;ERROR 72
1220
1221  002154  026351                        EM72        ;DRIVE BUS ERROR; 'SET CYLINDER' AND CYLINDER ADDRESS
1222                                                      ;SHOULD BE ON THE BUS
1223  002156  031141                        DH72
1224  002160  031606                        DT72
1225  002162  031672                        DF30
1226
1227                                ;ERROR 73
1228
1229  002164  026464                        EM73        ;DRIVE BUS ERROR: ONLY 'RESET HEAD' SHOULD BE ON THE BUS
1230  002166  031216                        DH73
1231  002170  031622                        DT73
1232  002172  031672                        DF30
1233
1234                                ;ERROR 74
1235
1236  002174  026554                        EM74        ;DRIVE BUS ERROR: 'SET HEAD' & HEAD ADDRESS SHOULD BE ON THE BUS
1237  002176  031275                        DH74
1238  002200  031636                        DT74
1239  002202  031672                        DF30
1240
1241                                ;ERROR 75
1242
1243  002204  026654                        EM75        ;DRIVE BUS ERROR: ONLY 'SEEK START' SHOULD BE ON THE BUS
1244
1245  002206  031216                        DH73
1246  002210  031622                        DT73
1247  002212  031672                        DF30
1248
1249                                ;ERROR 76
1250
```

```
1251  002214  026744                        EM76            ;DRIVE BUS ERROR: 'RESTORE' AND 'CONTROL' SHOULD BE ON THE BUS
1252  002216  031216                        DH73
1253  002220  031622                        DT73
1254  002222  031672                        DF30
1255
1256                                  ;ERROR 77
1257
1258  002224  027042                        EM77            ;DRIVE BUS ERROR: 'CONTROL', 'SELECT HEAD', & 'READ'
1259                                                        ;SHOULD BE ON THE BUS
1260  002226  031216                        DH73
1261  002230  031622                        DT73
1262  002232  031672                        DF30
1263
1264                                  ;ERROR 100
1265
1266  002234  027156                        EM100           ;DRIVE BUS ERROR; 'CONTROL', SELECT HEAD', 'ERASE',
1267                                                        ;AND 'WRITE' SHOULD BE ON THE BUS
1268  002236  031216                        DH73
1269  002240  031622                        DT73
1270  002242  031672                        DF30
1271
1272                                  ;ERROR 101
1273
1274  002244  027303                        EM101           ;'READY' DIDN'T CLEAR WHEN 'GO' WAS SET
1275  002246  030520                        DH52
1276  002250  031502                        DT52
1277  002252  031656                        DF1
1278
1279                                  ;ERROR 102
1280
1281  002254  027352                        EM102           ;CAN'T SET 'SUOL'
1282  002256  030035                        DH7
1283  002260  031370                        DT7
1284  002262  031656                        DF1
1285
1286                                  ;ERROR 103
1287
1288  002264  027373                        EM103           ;CAN'T CLEAR 'SUOL'
1289  002266  030035                        DH7
1290  002270  031370                        DT7
1291  002272  031656                        DF1
1292
1293                                  ;ERROR 104
1294
1295  002274  027416                        EM104           ;CAN'T SET 'SUSU'
1296  002276  030035                        DH7
1297  002300  031370                        DT7
1298  002302  031656                        DF1
1299
1300                                  ;ERROR 105
1301
1302  002304  027437                        EM105           ;CAN'T CLEAR 'SUSU'
1303  002306  030035                        DH7
1304  002310  031370                        DT7
1305  002312  031656                        DF1
1306
```

```
                                        ;ERROR 106
                                                EM106           ;'IDE' NOT SET AFTER INTERRUPT
002314  027462                                  DH52
002316  030520                                  DT52
002320  031502                                  DF1
002322  031656
                                        ;ERROR 107
                                                EM107           ;ATTN INTERRUPT OCCURED WITH NO ATTN BITS SET
002324  027520                                  DH11
002326  030062                                  DT11
002330  031376                                  DF2
002332  031662
                                        ;ERROR 110
                                                EM110           ;SILO NOT FULL, 'INPUT READY' NOT SET
002334  027575                                  DH65
002336  030751                                  DT65
002340  031546                                  DF1
002342  031656
                                        ;ERROR 111
                                                EM111           ;SILO NOT EMPTY, 'OUTPUT READY' NOT SET
002344  027642                                  DH65
002346  030751                                  DT65
002350  031546                                  DF1
002352  031656
                                        ;ERROR 112
                                                EM112           ;'INIT' COMMAND SET 'PROG' ERROR
002354  027711                                  DH24
002356  030164                                  DT24
002360  031420                                  DF20
002362  031666


                                        .SBTTL  START OF PROGRAM

002364  005037  001204                  START:  CLR     BUSADR          ;CLEAR THE ADDRESS CHANGE FLAG
002370  000403                                  BR      START2          ;GO TO SETUP
002372  012737  177777  001204          START1: MOV     #-1,BUSADR      ;SET ADDRESS CHANGE FLAG
002400  000005                          START2: RESET                   ;CLEAR THE BUS
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002402  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002406  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002410  022706  001140                          CMP     #SWR,R6 ;;DONE?
002414  001374                                  BNE     .-6             ;;LOOP BACK IF NO
002416  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
002422  012737  020734  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002430  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002436  012737  016404  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002444  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002452  012737  021310  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002460  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
```

```
1363  002466  013737  016320  016312          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1364  002474  005037  001162                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1365  002500  005037  001164                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1366  002504  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1367  002512  012737  002512  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1368  002520  012737  002520  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1369                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1370                                  ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1371  002526  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1372  002532  012737  002566  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1373  002540  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1374  002546  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1375  002554  022777  177777  176356          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1376  002562  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1377                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1378  002564  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1379  002566  012716  002574          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1380  002572  000002                          RTI
1381  002574  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1382  002602  012737  000174  001142          MOV     #DISPREG,DISPLAY
1383  002610  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1384
1385  002614  005037  001100                  CLR     $PASS           ;CLEAR THE PASS COUNT
1386  002620  005227  177777                  INC     #-1             ;FIRST START ?
1387  002624  001042                          BNE     1$              ;BR IF NOT
1388  002626  103441                          BCS     1$              ;BR IF NOT
1389  002630  104401  002636                  TYPE    ,68$            ;;TYPE ASCIZ STRING
1390  002634  000431                          BR      67$             ;;GET OVER THE ASCIZ
1391                                  ;;68$:  .ASCIZ  <15><12><12>/MAINDEC-11-DZRPW-B, RP11-E DISKLESS LOGIC TEST/
1392  002720                          67$:
1393  002720  005737  000042                  TST     @#42            ;UNDER MONITOR CONTROL ?
1394  002724  001002                          BNE     1$              ;BR IF YES
1395  002726  104401  022115                  TYPE    ,DRVOL          ;TYPE ONLINE MESSAGE
1396  002732                          1$:
1397                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1398  002732  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1399  002736  001006                          BNE     69$             ;;BRANCH IF YES
1400  002740  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1401  002746  001005                          BNE     70$             ;;BRANCH IF NO
1402  002750  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1403  002752  000403                          BR      70$
1404  002754  112737  000001  001134  69$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1405  002762                          70$:
1406  002762  004737  021376                  JSR     PC,GETADR       ;CHECK THE BUS ADDRESS
1407  002766  013777  001216  176220          MOV     RPPRIO,@RPVEC+2 ;SETUP RP11 PRIORITY
1408  002774  104401  001173          RSTART: TYPE    ,$CRLF          ;CR-LF
1409  003000  005037  000000                  CLR     @#0             ;CLEAR LOCATION 0
1410  003004  005077  176202                  CLR     @RPVEC          ;SETUP VECTOR TO LOCATION 0
1411  003010  013704  001210                  MOV     RPADR,R4        ;LOAD RP11 ADDRESS POINTER
1412  003014  012764  000000  000004          MOV     #0,RPCS(R4)     ;SELECT DRIVE 0
1413  003022  032764  020000  000000          BIT     #BIT13,RPDS(R4) ;DRIVE 0 AN RP03 ?
1414  003030  001007                          BNE     1$              ;BR IF NOT
1415  003032  012737  000312  001176          MOV     #202.,MAXCYL    ;SET MAXIMUM CYLINDER ADDRESS TO 202
1416  003040  012737  000400  001200          MOV     #256.,MAXPAT    ;CYLINDER BITS TO BE TESTED
1417  003046  000406                          BR      2$              ;CONTINUE
1418  003050  012737  000625  001176  1$:     MOV     #405.,MAXCYL    ;SET MAXIMUM CYLINDER ADDRESS TO 405
```

```
1419  003056  012737  001000  001200          MOV     #512.,MAXPAT    ;CYLINDER BITS TO BE TESTED
1420  003064  012737  003064  001110  2$:     MOV     #.,$LPERR       ;LOAD LOOP ON ERROR ADDRESS
1421  003072  012737  003072  001106          MOV     #.,$LPADR       ;LOAD LOOP ON TEST ADDRESS
1422  003100  012737  000340  177776          MOV     #PR7,PSW        ;SET PROCESSOR PRIORITY TO MAXIMUM
1423  003106  005037  001102                  CLR     $TSTNM          ;CLEAR TEST NUMBER
1424  003112  005037  001104                  CLR     $ICNT           ;CLEAR ITERATION COUNT
1425  003116  000137  003122                  JMP     TST0            ;START TEST 0
1426
1427
1428                                          .SBTTL  #### START OF TESTS ####
1429
1430
1431                                          ;;*****************************************************************
1432                                          ;*TEST 0        CONTROLLER RESET TEST
1433
1434                                          ;*TEST THAT 'RESET' CLEARS THE CONTROLLER.
1435
1436                                          ;;*****************************************************************
1437  003122                                  TST0:
1438  003122  000240                          NOP
1439  003124  012737  003152  001106          MOV     #RESRP,$LPADR   ;LOAD LOOP ON TEST ADDRESS
1440  003132  012737  003152  001110          MOV     #RESRP,$LPERR   ;LOAD ERROR LOOP ADDRESS
1441  003140  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
1442  003144  012737  000002  001162          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1443  003152  012706  001100          RESRP:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
1444  003156  012764  176656  000002          MOV     #176656,RPER(R4)  ;LOAD RPER
1445  003164  012764  037576  000004          MOV     #37576,RPCS(R4) ;LOAD RPCS
1446  003172  012764  177777  000006          MOV     #-1,RPWC(R4)    ;LOAD RPWC
1447  003200  012764  177777  000010          MOV     #-1,RPBA(R4)    ;LOAD RPBA
1448  003206  012764  000777  000012          MOV     #777,RPCA(R4)   ;LOAD RPCA
1449  003214  012764  017417  000014          MOV     #17417,RPDA(R4) ;LOAD RPDA
1450  003222  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1451  003226                          1$:
1452  003226  012737  000002  001122          MOV     #RPER,$BDADR    ;REGISTER ADDRESS INDEX
1453  003234  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1454  003240  012737  176656  001124          MOV     #176656,$GDDAT  ;GOOD DATA
1455  003246  013737  001222  001126          MOV     $RPER,$BDDAT    ;CONTENTS TO BE CHECKED
1456  003254  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1457  003262  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPER LOAD PROPERLY ?
1458  003270  001401                          BEQ     2$              ;BR IF IT DID
1459  003272  104002                          ERROR   2               ;COULD NOT LOAD RPER CORRECTLY
1460  003274                          2$:
1461  003274  012737  000004  001122          MOV     #RPCS,$BDADR    ;REGISTER ADDRESS INDEX
1462  003302  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1463  003306  012737  177776  001124          MOV     #177776,$GDDAT  ;GOOD DATA
1464  003314  013737  001224  001126          MOV     $RPCS,$BDDAT    ;CONTENTS TO BE CHECKED
1465  003322  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1466  003330  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPCS LOAD PROPERLY ?
1467  003336  001401                          BEQ     3$              ;BR IF IT DID
1468  003340  104002                          ERROR   2               ;COULD NOT LOAD RPCS CORRECTLY
1469  003342                          3$:
1470  003342  012737  000006  001122          MOV     #RPWC,$BDADR    ;REGISTER ADDRESS INDEX
1471  003350  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1472  003354  012737  177777  001124          MOV     #-1,$GDDAT      ;GOOD DATA
1473  003362  013737  001226  001126          MOV     $RPWC,$BDDAT    ;CONTENTS TO BE CHECKED
1474  003370  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
```

```
1475  003376  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPWC LOAD PROPERLY ?
1476  003404  001401                          BEQ     4$              ;BR IF IT DID
1477  003406  104002                          ERROR   2               ;COULD NOT LOAD RPWC CORRECTLY
1478  003410                          4$:
1479  003410  012737  000010  001122          MOV     #RPBA,$BDADR    ;REGISTER ADDRESS INDEX
1480  003416  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1481  003422  012737  177777  001124          MOV     #-1,$GDDAT      ;GOOD DATA
1482  003430  013737  001230  001126          MOV     $RPBA,$BDDAT    ;CONTENTS TO BE CHECKED
1483  003436  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1484  003444  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPBA LOAD PROPERLY ?
1485  003452  001401                          BEQ     5$              ;BR IF IT DID
1486  003454  104002                          ERROR   2               ;COULD NOT LOAD RPBA CORRECTLY
1487  003456                          5$:
1488  003456  012737  000012  001122          MOV     #RPCA,$BDADR    ;REGISTER ADDRESS INDEX
1489  003464  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1490  003470  012737  000777  001124          MOV     #777,$GDDAT     ;GOOD DATA
1491  003476  013737  001232  001126          MOV     $RPCA,$BDDAT    ;CONTENTS TO BE CHECKED
1492  003504  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1493  003512  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPCA LOAD PROPERLY ?
1494  003520  001401                          BEQ     6$              ;BR IF IT DID
1495  003522  104002                          ERROR   2               ;COULD NOT LOAD RPCA CORRECTLY
1496  003524                          6$:
1497  003524  012737  000014  001122          MOV     #RPDA,$BDADR    ;REGISTER ADDRESS INDEX
1498  003532  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1499  003536  012737  017417  001124          MOV     #17417,$GDDAT   ;GOOD DATA
1500  003544  013737  001234  001126          MOV     $RPDA,$BDDAT    ;CONTENTS TO BE CHECKED
1501  003552  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1502  003560  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPDA LOAD PROPERLY ?
1503  003566  001401                          BEQ     7$              ;BR IF IT DID
1504  003570  104002                          ERROR   2               ;COULD NOT LOAD RPDA CORRECTLY
1505  003572  000005                  7$:     RESET                   ;CLEAR THE RP11E
1506  003574  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
1507  003600  005037  001124                  CLR     $GDDAT          ;EXPECTED VALUE
1508  003604  012737  000002  001122          MOV     #RPER,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1509  003612  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1510  003616  013737  001222  001126          MOV     $RPER,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1511  003624  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPER CLEAR ?
1512  003632  001401                          BEQ     .+4             ;BR IF IT IS
1513  003634  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1514  003636  012737  000006  001122          MOV     #RPWC,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1515  003644  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1516  003650  013737  001226  001126          MOV     $RPWC,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1517  003656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPWC CLEAR ?
1518  003664  001401                          BEQ     .+4             ;BR IF IT IS
1519  003666  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1520  003670  012737  000010  001122          MOV     #RPBA,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1521  003676  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1522  003702  013737  001230  001126          MOV     $RPBA,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1523  003710  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPBA CLEAR ?
1524  003716  001401                          BEQ     .+4             ;BR IF IT IS
1525  003720  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1526  003722  012737  000012  001122          MOV     #RPCA,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1527  003730  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1528  003734  013737  001232  001126          MOV     $RPCA,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1529  003742  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPCA CLEAR ?
1530  003750  001401                          BEQ     .+4             ;BR IF IT IS
```

```
1531  003752  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1532  003754  012737  000016  001122          MOV     #RPM1,$BOADR    ;REGISTER ADDRESS INDEX VALUE
1533  003762  060437  001122                  ADD     R4,$BOADR       ;FORM REGISTER ADDRESS
1534  003766  013737  001236  001126          MOV     $RPM1,$BODAT    ;REGISTER VALUE FOR TYPEOUT
1535  003774  042737  004000  001126          BIC     #4000,$BODAT    ;CLEAR THE INPUT READY BIT
1536  004002  023737  001124  001126          CMP     $GODAT,$BODAT   ;IS RPM1 CLEAR ?
1537  004010  001401                          BEQ     .+4             ;BR IF IT IS
1538  004012  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1539  004014  012737  000014  001122          MOV     #RPDA,$BOADR    ;REGISTER ADDRESS INDEX VALUE
1540  004022  060437  001122                  ADD     R4,$BOADR       ;FORM REGISTER ADDRESS
1541  004026  013737  001234  001126          MOV     $RPDA,$BODAT    ;REGISTER CONTENTS FOR TYPEOUT
1542  004034  042737  000360  001126          BIC     #360,$BODAT     ;CLEAR THE 'SOT' BITS
1543  004042  023737  001124  001126          CMP     $GODAT,$BODAT   ;IS REGISTER RPDA CLEAR ?
1544  004050  001401                          BEQ     .+4             ;BR IF IT IS
1545  004052  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1546  004054  012737  000200  001124          MOV     #200,$GODAT     ;EXPECTED VALUE
1547  004062  013737  001224  001126          MOV     $RPCS,$BODAT    ;REGISTER CONTENTS FOR TYPEOUT
1548  004070  023737  001124  001126          CMP     $GODAT,$BODAT   ;CONTENTS CORRECT ?
1549  004076  001401                          BEQ     B$              ;BR IF CORRECT
1550  004100  104003                          ERROR   3               ;RESET DIDN'T CLEAR INDICATED REGISTER
1551  004102  000004                  B$:     SCOPE                   ;LOOP ?
1552
1553                                  ;;*****************************************************************
1554                                  ;*TEST 1        TEST SETTING 'SUOL' (SELECTED UNIT ONLINE)
1555
1556                                  ;;*****************************************************************
1557  004104                          TST1:
1558  004104  000240                          NOP
1559  004106  012737  004130  001106          MOV     #SETOL,$LPADR   ;LOAD LOOP ON TEST ADDRESS
1560  004114  012737  004130  001110          MOV     #SETOL,$LPERR   ;LOAD ERROR LOOP ADDRESS
1561  004122  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
1562  004126  000005                          RESET                   ;CLEAR THE CONTROLLER
1563  004130  012706  001100          SETOL:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
1564  004134  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1565  004140  052764  020000  000022          BIS     #BIT13,RPM3(R4) ;SET MAINT UNIT ONLINE
1566  004146  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
1567  004152  032737  040000  001220          BIT     #BIT14,$RPDS    ;DID SELECTED UNIT ONLINE SET ?
1568  004160  001002                          BNE     1$              ;BRANCH IF SET
1569  004162  104102                          ERROR   102             ;SELECTED UNIT ONLINE NOT SET
1570  004164  000412                          BR      2$              ;BYPASS REST OF CHECKS
1571  004166  042764  020000  000022  1$:     BIC     #BIT13,RPM3(R4) ;CLEAR MAINT UNIT ONLINE
1572  004174  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
1573  004200  032737  040000  001220          BIT     #BIT14,$RPDS    ;DID SELECTED UNIT ONLINE CLEAR?
1574  004206  001401                          BEQ     2$
1575  004210  104103                          ERROR   103             ;SELECTED UNIT ONLINE DID NOT CLEAR
1576  004212  000004                  2$:     SCOPE                   ;LOOP ?
1577  004214  052764  020000  000022          BIS     #BIT13,RPM3(R4) ;SET MAINT DRIVE ONLINE
1578
1579
1580                                  ;;*****************************************************************
1581                                  ;*TEST 2        CONTROLLER CLEAR TEST
1582
1583                                  ;*TEST THE ABILITY OF BIT 0 OF RPCS TO CLEAR THE CONTROLLER
1584                                  ;*WHEN THE COMMAND BITS ARE 0.
1585
1586                                  ;;*****************************************************************
```

```
1587  004222                                  TST2:
1588  004222  000240                                  NOP
1589  004224  012737  004252  001106                  MOV     #CLEARP,$LPADR  ;LOAD LOOP ON TEST ADDRESS
1590  004232  012737  004252  001110                  MOV     #CLEARP,$LPERR  ;LOAD ERROR LOOP ADDRESS
1591  004240  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
1592  004244  012737  000002  001162                  MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1593  004252  000005                          CLEARP: RESET                   ;CLEAR THE BUS
1594  004254  012706  001100                          MOV     #STACK,SP       ;SETUP THE STACK POINTER
1595  004260  012764  176656  000002                  MOV     #176656,RPER(R4)  ;LOAD RPER
1596  004266  012764  037576  000004                  MOV     #37576,RPCS(R4) ;LOAD RPCS
1597  004274  012764  177777  000006                  MOV     #-1,RPWC(R4)    ;LOAD RPWC
1598  004302  012764  177777  000010                  MOV     #-1,RPBA(R4)    ;LOAD RPBA
1599  004310  012764  000777  000012                  MOV     #777,RPCA(R4)   ;LOAD RPCA
1600  004316  012764  017417  000014                  MOV     #17417,RPDA(R4) ;LOAD RPDA
1601  004324  004737  022014                          JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1602  004330                                  1$:
1603  004330  012737  000002  001122                  MOV     #RPER,$BDADR    ;REGISTER ADDRESS INDEX
1604  004336  060437  001122                          ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1605  004342  012737  176656  001124                  MOV     #176656,$GDDAT  ;GOOD DATA
1606  004350  013737  001222  001126                  MOV     $RPER,$BDDAT    ;CONTENTS TO BE CHECKED
1607  004356  053737  001126  001124                  BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1608  004364  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;DID RPER LOAD PROPERLY ?
1609  004372  001401                                  BEQ     2$              ;BR IF IT DID
1610  004374  104002                                  ERROR   2               ;COULD NOT LOAD RPER CORRECTLY
1611  004376                                  2$:
1612  004376  012737  000004  001122                  MOV     #RPCS,$BDADR    ;REGISTER ADDRESS INDEX
1613  004404  060437  001122                          ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1614  004410  012737  177776  001124                  MOV     #177776,$GDDAT  ;GOOD DATA
1615  004416  013737  001224  001126                  MOV     $RPCS,$BDDAT    ;CONTENTS TO BE CHECKED
1616  004424  053737  001126  001124                  BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1617  004432  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;DID RPCS LOAD PROPERLY ?
1618  004440  001401                                  BEQ     3$              ;BR IF IT DID
1619  004442  104002                                  ERROR   2               ;COULD NOT LOAD RPCS CORRECTLY
1620  004444                                  3$:
1621  004444  012737  000006  001122                  MOV     #RPWC,$BDADR    ;REGISTER ADDRESS INDEX
1622  004452  060437  001122                          ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1623  004456  012737  177777  001124                  MOV     #-1,$GDDAT      ;GOOD DATA
1624  004464  013737  001226  001126                  MOV     $RPWC,$BDDAT    ;CONTENTS TO BE CHECKED
1625  004472  053737  001126  001124                  BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1626  004500  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;DID RPWC LOAD PROPERLY ?
1627  004506  001401                                  BEQ     4$              ;BR IF IT DID
1628  004510  104002                                  ERROR   2               ;COULD NOT LOAD RPWC CORRECTLY
1629  004512                                  4$:
1630  004512  012737  000010  001122                  MOV     #RPBA,$BDADR    ;REGISTER ADDRESS INDEX
1631  004520  060437  001122                          ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1632  004524  012737  177777  001124                  MOV     #-1,$GDDAT      ;GOOD DATA
1633  004532  013737  001230  001126                  MOV     $RPBA,$BDDAT    ;CONTENTS TO BE CHECKED
1634  004540  053737  001126  001124                  BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1635  004546  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;DID RPBA LOAD PROPERLY ?
1636  004554  001401                                  BEQ     5$              ;BR IF IT DID
1637  004556  104002                                  ERROR   2               ;COULD NOT LOAD RPBA CORRECTLY
1638  004560                                  5$:
1639  004560  012737  000012  001122                  MOV     #RPCA,$BDADR    ;REGISTER ADDRESS INDEX
1640  004566  060437  001122                          ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1641  004572  012737  000777  001124                  MOV     #777,$GDDAT     ;GOOD DATA
1642  004600  013737  001232  001126                  MOV     $RPCA,$BDDAT    ;CONTENTS TO BE CHECKED
```

```
1643  004606  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1644  004614  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPCA LOAD PROPERLY ?
1645  004622  001401                          BEQ     6$              ;BR IF IT DID
1646  004624  104002                          ERROR   2               ;COULD NOT LOAD RPCA CORRECTLY
1647  004626                          6$:
1648  004626  012737  000014  001122          MOV     #RPDA,$BDADR    ;REGISTER ADDRESS INDEX
1649  004634  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1650  004640  012737  017417  001124          MOV     #17417,$GDDAT   ;GOOD DATA
1651  004646  013737  001234  001126          MOV     $RPDA,$BDDAT    ;CONTENTS TO BE CHECKED
1652  004654  053737  001126  001124          BIS     $BDDAT,$GDDAT   ;SET UNUSED BITS FOR TYPEOUT
1653  004662  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID RPDA LOAD PROPERLY ?
1654  004670  001401                          BEQ     7$              ;BR IF IT DID
1655  004672  104002                          ERROR   2               ;COULD NOT LOAD RPDA CORRECTLY
1656  004674  012737  060001  001206  7$:     MOV     #BIT14!BIT13!BIT00,TPL
1657  004702  112767  000001  000004          MOVB    #1,RPCS(R4)     ;SET THE 'GO' BIT
1658  004710  004737  021756                  JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
1659  004714  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
1660  004720  032737  002000  001222          BIT     #BIT10,$RPER    ;DID 'PROG' SET WITH CLEAR ?
1661  004726  001401                          BEQ     9$              ;BR IF NOT
1662  004730  104112                          ERROR   112
1663  004732  005037  001124          9$:     CLR     $GDDAT          ;EXPECTED VALUE
1664  004736  012737  000002  001122          MOV     #RPER,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1665  004744  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1666  004750  013737  001222  001126          MOV     $RPER,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1667  004756  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPER CLEAR ?
1668  004764  001401                          BEQ     .+4             ;BR IF IT IS
1669  004766  104004                          ERROR   4               ;CONTROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1670  004770  012737  000006  001122          MOV     #RPWC,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1671  004776  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1672  005002  013737  001226  001126          MOV     $RPWC,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1673  005010  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPWC CLEAR ?
1674  005016  001401                          BEQ     .+4             ;BR IF IT IS
1675  005020  104004                          ERROR   4               ;CONTROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1676  005022  012737  000010  001122          MOV     #RPBA,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1677  005030  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1678  005034  013737  001230  001126          MOV     $RPBA,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1679  005042  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPBA CLEAR ?
1680  005050  001401                          BEQ     .+4             ;BR IF IT IS
1681  005052  104004                          ERROR   4               ;CONTROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1682  005054  012737  000012  001122          MOV     #RPCA,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1683  005062  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1684  005066  013737  001232  001126          MOV     $RPCA,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1685  005074  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPCA CLEAR ?
1686  005102  001401                          BEQ     .+4             ;BR IF IT IS
1687  005104  104004                          ERROR   4               ;CONTROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1688  005106  012737  000016  001122          MOV     #RPM1,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1689  005114  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1690  005120  013737  001236  001126          MOV     $RPM1,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1691  005126  042737  004000  001126          BIC     #4000,$BDDAT    ;CLEAR THE INPUT READY BIT
1692  005134  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS RPM1 CLEAR ?
1693  005142  001401                          BEQ     .+4             ;BR IF IT IS
1694  005144  104004                          ERROR   4               ;CONTROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1695  005146  012737  000014  001122          MOV     #RPDA,$BDADR    ;REGISTER ADDRESS INDEX VALUE
1696  005154  060437  001122                  ADD     R4,$BDADR       ;FORM REGISTER ADDRESS
1697  005160  013737  001234  001126          MOV     $RPDA,$BDDAT    ;REGISTER VALUE FOR TYPEOUT
1698  005166  042737  000360  001126          BIC     #360,$BDDAT     ;CLEAR THE 'SOT' BITS
```

```
1699  005174  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REGISTER RPDA CLEAR ?
1700  005202  001401                          BEQ     .+4             ;BR IF IT IS
1701  005204  104004                          ERROR   4               ;CONROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1702  005206  012737  000200  001124          MOV     #200,$GDDAT     ;EXPECTED VALUE
1703  005214  013737  001224  001126          MOV     $RPCS,$BDDAT    ;REGISTER CONTENTS FOR TYPEOUT
1704  005222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CONTENTS CORRECT ?
1705  005230  001401                          BEQ     8$              ;BR IF CORRECT
1706  005232  104004                          ERROR   4               ;CONTROLLER CLEAR DIDN'T CLEAR INDICATED REGISTER
1707  005234  052764  020000  000022  8$:     BIS     #BIT13,RPM3(R4) ;SET MAINTENANCE DRIVE ONLINE
1708  005242  000000                          SCOPE                   ;LOOP ?
1709
1710                                  ;*THE FOLLOWING 6 TESTS TEST LOADING AND READING OF ALL POSSIBLE
1711                                  ;*BITS IN REGISTERS RPER, RPCS, RPWC, RPBA, RPCA AND
1712                                  ;*RPDA USING FLOATING 1'S AND FLOATING 0'S PATTERNS.
1713
1714
1715                                  ;;*****************************************************************
1716                                  ;*TEST 3         TEST BITS IN 'RPER'
1717
1718                                  ;;*****************************************************************
1719  005244                          TST3:
1720  005244  000240                          NOP
1721  005246  012737  005274  001106          MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
1722  005254  012737  005324  001110          MOV     #ERBT,$LPERR    ;LOAD ERROR LOOP ADDRESS
1723  005262  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
1724  005266  012737  000012  001162          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1725  005274                          1$:
1726  005274  012706  001100                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
1727  005300  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1728  005304  010437  001122                  MOV     R4,$BDADR       ;RP11 ADDRESS
1729  005310  062737  000002  001122          ADD     #RPER,$BDADR    ;FORM ADDRESS OF REGISTER BEING CHECKED
1730  005316  012700  000001                  MOV     #1,R0           ;BEGINNING TEST PATTERN
1731  005322  005001                          CLR     R1              ;FLOATING 1'S INDICATOR
1732  005324  010037  001124          ERBT:   MOV     R0,$GDDAT       ;REFERENCE PATTERN
1733  005330  042737  001121  001124          BIC     #^C176656,$GDDAT        ;CLEAR UNUSED BITS
1734  005336  013764  001124  000002          MOV     $GDDAT,RPER(R4)  ;PUT PATTERN IN RPER
1735  005344  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1736  005350  013737  001222  001126          MOV     $RPER,$BDDAT    ;VALUE FOR TYPEOUT IN CASE OF ERROR
1737  005356  013737  001222  001160          MOV     $RPER,$TMP0     ;WORKING LOCATION FOR CHECKING
1738  005364  042737  001121  001160          BIC     #^C176656,$TMP0  ;CLEAR UNUSED BITS
1739  005372  023737  001124  001160          CMP     $GDDAT,$TMP0    ;REGISTER LOADED OK ?
1740  005400  001403                          BEQ     2$              ;BR IF OK
1741  005402  104005                          ERROR   5               ;FAILURE IN FLOATING 1'S & 0'S TEST
1742  005404  005064  000002                  CLR     RPER(R4)        ;CLEAR THE REGISTER
1743  005410  005100                  2$:     COM     R0              ;COMPLIMENT TEST PATTERN
1744  005412  005101                          COM     R1              ;COMPLIMENT PASS INDICATOR
1745  005414  001343                          BNE     ERBT            ;BR IF JUST FINISHED FLOATING 1'S PASS
1746  005416  006300                          ASL     R0              ;SHIFT THE TEST PATTERN
1747  005420  103341                          BCC     ERBT            ;BR IF MORE TO DO
1748  005422  005064  000002                  CLR     RPER(R4)        ;CLEAR THE REGISTER
1749  005426  000004                          SCOPE                   ;LOOP ?
1750
1751                                  ;;*****************************************************************
1752                                  ;*TEST 4         TEST BITS IN 'RPCS'
1753
1754                                  ;;*****************************************************************
```

```
1755  005430                                  TST4:
1756  005430  000240                                  NOP
1757  005432  012737  005460  001106                  MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
1758  005440  012737  005510  001110                  MOV     #CSBT,$LPERR    ;LOAD ERROR LOOP ADDRESS
1759  005446  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
1760  005452  012737  000012  001162                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1761  005460                                  1$:
1762  005460  012706  001100                          MOV     #STACK,SP       ;SETUP THE STACK POINTER
1763  005464  004737  021706                          JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1764  005470  010437  001122                          MOV     R4,$BOADR       ;RP11 ADDRESS
1765  005474  062737  000004  001122                  ADD     #RPCS,$BOADR    ;FORM ADDRESS OF REGISTER BEING CHECKED
1766  005502  012700  000001                          MOV     #1,R0           ;BEGINNING TEST PATTERN
1767  005506  005001                                  CLR     R1              ;FLOATING 1'S INDICATOR
1768  005510  010037  001124                  CSBT:   MOV     R0,$GDDAT       ;REFERENCE PATTERN
1769  005514  042737  140201  001124                  BIC     #^C37576,$GDDAT ;CLEAR UNUSED BITS
1770  005522  013764  001124  000004                  MOV     $GDDAT,RPCS(R4) ;PUT PATTERN IN RPCS
1771  005530  004737  022014                          JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1772  005534  013737  001224  001126                  MOV     $RPCS,$BDDAT    ;VALUE FOR TYPEOUT IN CASE OF ERROR
1773  005542  013737  001224  001160                  MOV     $RPCS,$TMP0     ;WORKING LOCATION FOR CHECKING
1774  005550  042737  140201  001160                  BIC     #^C37576,$TMP0  ;CLEAR UNUSED BITS
1775  005556  023737  001124  001160                  CMP     $GDDAT,$TMP0    ;REGISTER LOADED OK ?
1776  005564  001403                                  BEQ     2$              ;BR IF OK
1777  005566  104205                                  ERROR   5               ;FAILURE IN FLOATING 1'S & 0'S TEST
1778  005570  005064  000004                          CLR     RPCS(R4)        ;CLEAR THE REGISTER
1779  005574  005100                          2$:     COM     R0              ;COMPLIMENT TEST PATTERN
1780  005576  005101                                  COM     R1              ;COMPLIMENT PASS INDICATOR
1781  005600  001343                                  BNE     CSBT            ;BR IF JUST FINISHED FLOATING 1'S PASS
1782  005602  006300                                  ASL     R0              ;SHIFT THE TEST PATTERN
1783  005604  103341                                  BCC     CSBT            ;BR IF MORE TO DO
1784  005606  005064  000004                          CLR     RPCS(R4)        ;CLEAR THE REGISTER
1785  005612  000004                                  SCOPE                   ;LOOP ?
1786
1787                                  ;;*****************************************************************
1788                                  ;*TEST 5         TEST BITS IN 'RPWC'
1789
1790                                  ;;*****************************************************************
1791  005614                                  TST5:
1792  005614  000240                                  NOP
1793  005616  012737  005644  001106                  MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
1794  005624  012737  005674  001110                  MOV     #WCBT,$LPERR    ;LOAD ERROR LOOP ADDRESS
1795  005632  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
1796  005636  012737  000012  001162                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1797  005644                                  1$:
1798  005644  012706  001100                          MOV     #STACK,SP       ;SETUP THE STACK POINTER
1799  005650  004737  021706                          JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1800  005654  010437  001122                          MOV     R4,$BOADR       ;RP11 ADDRESS
1801  005660  062737  000006  001122                  ADD     #RPWC,$BOADR    ;FORM ADDRESS OF REGISTER BEING CHECKED
1802  005666  012700  000001                          MOV     #1,R0           ;BEGINNING TEST PATTERN
1803  005672  005001                                  CLR     R1              ;FLOATING 1'S INDICATOR
1804  005674  010037  001124                  WCBT:   MOV     R0,$GDDAT       ;REFERENCE PATTERN
1805  005700  042737  000000  001124                  BIC     #^C177777,$GDDAT        ;CLEAR UNUSED BITS
1806  005706  013764  001124  000006                  MOV     $GDDAT,RPWC(R4) ;PUT PATTERN IN RPWC
1807  005714  004737  022014                          JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1808  005720  013737  001226  001126                  MOV     $RPWC,$BDDAT    ;VALUE FOR TYPEOUT IN CASE OF ERROR
1809  005726  013737  001226  001160                  MOV     $RPWC,$TMP0     ;WORKING LOCATION FOR CHECKING
1810  005734  042737  000000  001160                  BIC     #^C177777,$TMP0 ;CLEAR UNUSED BITS
```

```
 1811  005742  023737  001124  001160          CMP     $GDDAT,$TMPD    ;REGISTER LOADED OK ?
 1812  005750  001403                          BEQ     2$              ;BR IF OK
 1813  005752  104005                          ERROR   5               ;FAILURE IN FLOATING 1'S & 0'S TEST
 1814  005754  005064  000006                  CLR     RPWC(R4)        ;CLEAR THE REGISTER
 1815  005760  005100                  2$:     COM     R0              ;COMPLIMENT TEST PATTERN
 1816  005762  005101                          COM     R1              ;COMPLIMENT PASS INDICATOR
 1817  005764  001343                          BNE     WCBT            ;BR IF JUST FINISHED FLOATING 1'S PASS
 1818  005766  006300                          ASL     R0              ;SHIFT THE TEST PATTERN
 1819  005770  103341                          BCC     WCBT            ;BR IF MORE TO DO
 1820  005772  005064  000006                  CLR     RPWC(R4)        ;CLEAR THE REGISTER
 1821  005776  000004                          SCOPE                   ;LOOP ?
 1822
 1823                                  ;;**********************************************************************
 1824                                  ;*TEST 6         TEST BITS IN 'RPBA'
 1825
 1826                                  ;;**********************************************************************
 1827  006000                          TST6:
 1828  006000  000240                          NOP
 1829  006002  012737  006030  001106          MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
 1830  006010  012737  006060  001110          MOV     #BABT,$LPERR    ;LOAD ERROR LOOP ADDRESS
 1831  006016  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 1832  006022  012737  000012  001162          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 1833  006030                          1$:
 1834  006030  012706  001100                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
 1835  006034  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 1836  006040  010437  001122                  MOV     R4,$BOADR       ;RP11 ADDRESS
 1837  006044  062737  000010  001122          ADD     #RPBA,$BOADR    ;FORM ADDRESS OF REGISTER BEING CHECKED
 1838  006052  012700  000001                  MOV     #1,R0           ;BEGINNING TEST PATTERN
 1839  006056  005001                          CLR     R1              ;FLOATING 1'S INDICATOR
 1840  006060  010037  001124          BABT:   MOV     R0,$GDDAT       ;REFERENCE PATTERN
 1841  006064  042737  000000  001124          BIC     #^C177777,$GDDAT        ;CLEAR UNUSED BITS
 1842  006072  013764  001124  000010          MOV     $GDDAT,RPBA(R4) ;PUT PATTERN IN RPBA
 1843  006100  004737  022014                  JSR     PC,$SAVRP       ;SAVE THE RP11 REGISTERS
 1844  006104  013737  001230  001126          MOV     $RPBA,$EDDAT    ;VALUE FOR TYPEOUT IN CASE OF ERROR
 1845  006112  013737  001230  001160          MOV     $RPBA,$TMPD     ;WORKING LOCATION FOR CHECKING
 1846  006120  042737  000000  001160          BIC     #^C177777,$TMPD ;CLEAR UNUSED BITS
 1847  006126  023737  001124  001160          CMP     $GDDAT,$TMPD    ;REGISTER LOADED OK ?
 1848  006134  001403                          BEQ     2$              ;BR IF OK
 1849  006136  104005                          ERROR   5               ;FAILURE IN FLOATING 1'S & 0'S TEST
 1850  006140  005064  000010                  CLR     RPBA(R4)        ;CLEAR THE REGISTER
 1851  006144  005100                  2$:     COM     R0              ;COMPLIMENT TEST PATTERN
 1852  006146  005101                          COM     R1              ;COMPLIMENT PASS INDICATOR
 1853  006150  001343                          BNE     BABT            ;BR IF JUST FINISHED FLOATING 1'S PASS
 1854  006152  006300                          ASL     R0              ;SHIFT THE TEST PATTERN
 1855  006154  103341                          BCC     BABT            ;BR IF MORE TO DO
 1856  006156  005064  000010                  CLR     RPBA(R4)        ;CLEAR THE REGISTER
 1857  006162  000004                          SCOPE                   ;LOOP ?
 1858
 1859                                  ;;**********************************************************************
 1860                                  ;*TEST 7         TEST BITS IN 'RPCA'
 1861
 1862                                  ;;**********************************************************************
 1863  006164                          TST7:
 1864  006164  000240                          NOP
 1865  006166  012737  006214  001106          MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
 1866  006174  012737  006244  001110          MOV     #CABT,$LPERR    ;LOAD ERROR LOOP ADDRESS
```

```
1867  006202  013704  001210                  MOV     RPADF,R4        ;RP11E BUS ADDRESS
1868  006206  012737  000012  001162          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1869  006214                          1$:
1870  006214  012706  001100                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
1871  006220  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1872  006224  010437  001122                  MOV     R4,$BDADR       ;RP11 ADDRESS
1873  006230  062737  000012  001122          ADD     #RPCA,$BDADR    ;FORM ADDRESS OF REGISTER BEING CHECKED
1874  006236  012700  000001                  MOV     #1,R0           ;BEGINNING TEST PATTERN
1875  006242  005001                          CLR     R1              ;FLOATING 1'S INDICATOR
1876  006244  010037  001124          CABT:   MOV     R0,$GDDAT       ;REFERENCE PATTERN
1877  006250  042737  177000  001124          BIC     #^C777,$GDDAT   ;CLEAR UNUSED BITS
1878  006256  013764  001124  000012          MOV     $GDDAT,RPCA(R4)  ;PUT PATTERN IN RPCA
1879  006264  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1880  006270  013737  001232  001126          MOV     $RPCA,$BDDAT    ;VALUE FOR TYPEOUT IN CASE OF ERROR
1881  006276  013737  001232  001160          MOV     $RPCA,$TMP0     ;WORKING LOCATION FOR CHECKING
1882  006304  042737  177000  001160          BIC     #^C777,$TMP0    ;CLEAR UNUSED BITS
1883  006312  023737  001124  001160          CMP     $GDDAT,$TMP0    ;REGISTER LOADED OK ?
1884  006320  001403                          BEQ     2$              ;BR IF OK
1885  006322  104005                          ERROR   5               ;FAILURE IN FLOATING 1'S & 0'S TEST
1886  006324  005064  000012                  CLR     RPCA(R4)        ;CLEAR THE REGISTER
1887  006330  005100                  2$:     COM     R0              ;COMPLIMENT TEST PATTERN
1888  006332  005101                          COM     R1              ;COMPLIMENT PASS INDICATOR
1889  006334  001343                          BNE     CABT            ;BR IF JUST FINISHED FLOATING 1'S PASS
1890  006336  006300                          ASL     R0              ;SHIFT THE TEST PATTERN
1891  006340  103341                          BCC     CABT            ;BR IF MORE TO DO
1892  006342  005064  000012                  CLR     RPCA(R4)        ;CLEAR THE REGISTER
1893  006346  000004                          SCOPE                   ;LOOP ?
1894
1895                                  ;;*****************************************************************
1896                                  ;*TEST 10       TEST BITS IN 'RPDA'
1897
1898                                  ;;*****************************************************************
1899  006350                          TST10:
1900  006350  000240                          NOP
1901  006352  012737  006400  001106          MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
1902  006360  012737  006430  001110          MOV     #CABT,$LPERR    ;LOAD ERROR LOOP ADDRESS
1903  006366  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
1904  006372  012737  000012  001162          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
1905  006400                          1$:
1906  006400  012706  001100                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
1907  006404  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1908  006410  010437  001122                  MOV     R4,$BDADR       ;RP11 ADDRESS
1909  006414  062737  000014  001122          ADD     #RPDA,$BDADR    ;FORM ADDRESS OF REGISTER BEING CHECKED
1910  006422  012700  000001                  MOV     #1,R0           ;BEGINNING TEST PATTERN
1911  006426  005001                          CLR     R1              ;FLOATING 1'S INDICATOR
1912  006430  010037  001124          DABT:   MOV     R0,$GDDAT       ;REFERENCE PATTERN
1913  006434  042737  160360  001124          BIC     #^C17417,$GDDAT ;CLEAR UNUSED BITS
1914  006442  013764  001124  000014          MOV     $GDDAT,RPDA(R4)  ;PUT PATTERN IN RPDA
1915  006450  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
1916  006454  013737  001234  001126          MOV     $RPDA,$BDDAT    ;VALUE FOR TYPEOUT IN CASE OF ERROR
1917  006462  013737  001234  001160          MOV     $RPDA,$TMP0     ;WORKING LOCATION FOR CHECKING
1918  006470  042737  160360  001160          BIC     #^C17417,$TMP0  ;CLEAR UNUSED BITS
1919  006476  023737  001124  001160          CMP     $GDDAT,$TMP0    ;REGISTER LOADED OK ?
1920  006504  001403                          BEQ     2$              ;BR IF OK
1921  006506  104005                          ERROR   5               ;FAILURE IN FLOATING 1'S & 0'S TEST
1922  006510  005064  000014                  CLR     RPDA(R4)        ;CLEAR THE REGISTER
```

```
1923  006514  005100                  25:     COM     R0              ;COMPLIMENT TEST PATTERN
1924  006516  005101                          COM     R1              ;COMPLIMENT PASS INDICATOR
1925  006520  001343                          BNE     DABT            ;BR IF JUST FINISHED FLOATING 1'S PASS
1926  006522  006300                          ASL     R0              ;SHIFT THE TEST PATTERN
1927  006524  103341                          BCC     DABT            ;BR IF MORE TO DO
1928  006526  005064  000014                  CLR     RPDA(R4)        ;CLEAR THE REGISTER
1929  006532  000004                          SCOPE                   ;LOOP ?
1930
1931                                  ;;*****************************************************************
1932                                  ;*TEST 11       'RPWC' RAPID ACCESS TEST
1933
1934                                  ;;*****************************************************************
1935  006534                          TST11:
1936  006534  000240                          NOP
1937  006536  012737  006564  001106          MOV     #1$,SLPADR      ;LOAD LOOP ON TEST ADDRESS
1938  006544  012737  006622  001110          MOV     #RAPWC,SLPERR   ;LOAD ERROR LOOP ADDRESS
1939  006552  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
1940  006556  012737  000002  001162          MOV     #2,$TIMES       ;DO 2 ITERATIONS
1941  006564  012706  001100          1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
1942  006570  012737  000006  001122          MOV     #RPWC,$BOADR    ;INDEX OF REGISTER TESTED
1943  006576  060437  001122                  ADD     R4,$BOADR       ;ADD THE BUS ADDRESS
1944  006602  013701  001122                  MOV     $BOADR,R1       ;COPY REGISTER ADDRESS
1945  006606  012737  006660  000004          MOV     #RAPWC1,@#ERRVEC  ;SETUP FOR END MEMORY TRAP
1946  006614  005000                          CLR     R0              ;SETUP START MEMORY LOCATION
1947  006616  004737  021706                  JSR     PC,CLRP         ;CLEAR THE RP11E
1948  006622  011011                  RAPWC:  MOV     (R0),(R1)       ;MOVE MEMORY TO REGISTER
1949  006624  011102                          MOV     (R1),R2         ;READ THE REGISTER
1950  006626  020220                          CMP     R2,(R0)+        ;TEST FOR PROPER VALUE IN REGISTER
1951  006630  001774                          BEQ     RAPWC           ;BR IF CORRECT
1952  006632  014037  001124                  MOV     -(R0),$GOODAT   ;MOVE GOOD DATA TO $GOODAT FOR TYPOUT
1953  006636  010237  001126                  MOV     R2,$BODAT       ;MOVE REGISTER DATA TO $BODAT FOR TYPEOUT
1954  006642  104006                          ERROR   6               ;REGISTER CONTENTS INCORRECT
1955  006644  032777  000200  172266          BIT     #SW07,@SWR      ;PRINT ANY MORE ERRORS ?
1956  006652  001002                          BNE     RAPWC1          ;BR IF NOT
1957  006654  005720                          TST     (R0)+           ;CONTINUE TEST
1958  006656  000761                          BR      RAPWC
1959  006660  012737  000006  000004  RAPWC1: MOV     #ERRVEC+2,@#ERRVEC  ;RESTORE TRAP CATCHER
1960  006666  012737  000340  177776          MOV     #PR7,@#PSW      ;RESTORE NO INTERUPTS
1961  006674  005000                          CLR     R0              ;CLEAR THE REGISTER
1962  006676  000004                          SCOPE                   ;LOOP ?
1963
1964                                  ;;*****************************************************************
1965                                  ;*TEST 12       'RPBA' RAPID ACCESS TEST
1966
1967                                  ;;*****************************************************************
1968  006700                          TST12:
1969  006700  000240                          NOP
1970  006702  012737  006730  001106          MOV     #1$,SLPADR      ;LOAD LOOP ON TEST ADDRESS
1971  006710  012737  006766  001110          MOV     #RAPBA,SLPERR   ;LOAD ERROR LOOP ADDRESS
1972  006716  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
1973  006722  012737  000002  001162          MOV     #2,$TIMES       ;DO 2 ITERATIONS
1974  006730  012706  001100          1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
1975  006734  012737  000010  001122          MOV     #RPBA,$BOADR    ;INDEX OF REGISTER TESTED
1976  006742  060437  001122                  ADD     R4,$BOADR       ;ADD THE BUS ADDRESS
1977  006746  013701  001122                  MOV     $BOADR,R1       ;COPY REGISTER ADDRESS
1978  006752  012737  006764  000004          MOV     #RAPBA1,@#ERRVEC  ;SETUP FOR END MEMORY TRAP
```

```
 1979  006760  005000                          CLR     R0              ;SETUP START MEMORY LOCATION
 1980  006762  004737  021706                  JSR     PC,CLRP         ;CLEAR THE RP11E
 1981  006766  011011                  RAPBA:  MOV     (R0),(R1)       ;MOVE MEMORY TO REGISTER
 1982  006770  011102                          MOV     (R1),R2         ;READ THE REGISTER
 1983  006772  020220                          CMP     R2,(R0)+        ;TEST FOR PROPER VALUE IN REGISTER
 1984  006774  001774                          BEQ     RAPBA           ;BR IF CORRECT
 1985  006776  014037  001124                  MOV     -(R0),$GDDAT    ;MOVE GOOD DATA TO $GDDAT FOR TYPOUT
 1986  007002  010237  001126                  MOV     R2,$BDDAT       ;MOVE REGISTER DATA TO $BDDAT FOR TYPEOUT
 1987  007006  104006                          ERROR   6               ;REGISTER CONTENTS INCORRECT
 1988  007010  032777  000200  172122          BIT     @SW07,@SWR      ;PRINT ANY MORE ERRORS ?
 1989  007016  001002                          BNE     RAPBA1          ;BR IF NOT
 1990  007020  005720                          TST     (R0)+           ;CONTINUE TEST
 1991  007022  000761                          BR      RAPBA
 1992  007024  012737  000006  000004  RAPBA1: MOV     #ERRVEC+2,@#ERRVEC  ;RESTORE TRAP CATCHER
 1993  007032  012737  000340  177776          MOV     #PR7,@#PSW      ;RESTORE NO INTERUPTS
 1994  007040  005000                          CLR     R0              ;CLEAR THE REGISTER
 1995  007042  000004                          SCOPE                   ;LOOP ?
 1996
 1997                                  ;;**************************************************************
 1998                                  ;*TEST 13       TEST SETTING 'SURDY' (SELECTED UNIT READY)
 1999
 2000                                  ;;**************************************************************
 2001  007044                          TST13:
 2002  007044  000240                          NOP
 2003  007046  012737  007066  001106          MOV     #DSF1,$LPADR    ;LOAD LOOP ON TEST ADDRESS
 2004  007054  012737  007066  001110          MOV     #DSF1,$LPERR    ;LOAD ERROR LOOP ADDRESS
 2005  007062  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2006  007066  012706  001100          DSF1:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2007  007072  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2008  007076  052764  040000  000022          BIS     #BIT14,RPM3(R4) ;SET MAINT UNIT READY
 2009  007104  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2010  007110  032737  100000  001220          BIT     #BIT15,$RPDS    ;DID SELECTED UNIT READY SET?
 2011  007116  001002                          BNE     1$              ;BRANCH IF SET
 2012  007120  104007                          ERROR   7               ;SELECTED UNIT READY NOT SET
 2013  007122  000412                          BR      2$              ;BYPASS REST OF CHECKS
 2014  007124  042764  040000  000022  1$:     BIC     #BIT14,RPM3(R4) ;CLEAR MAINT UNIT READY
 2015  007132  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2016  007136  032737  100000  001220          BIT     #BIT15,$RPDS    ;DID SELECTED UNIT READY CLEAR?
 2017  007144  001401                          BEQ     2$
 2018  007146  104010                          ERROR   10              ;SELECTED UNIT READY DID NOT CLEAR
 2019  007150  000004                  2$:     SCOPE                   ;LOOP ?
 2020
 2021
 2022                                  ;;**************************************************************
 2023                                  ;*TEST 14       TEST SETTING OF 'SUSI' (SELECTED UNIT SEEK INCOMPLETE
 2024
 2025                                  ;;**************************************************************
 2026  007152                          TST14:
 2027  007152  000240                          NOP
 2028  007154  012737  007174  001106          MOV     #DSF4,$LPADR    ;LOAD LOOP ON TEST ADDRESS
 2029  007162  012737  007174  001110          MOV     #DSF4,$LPERR    ;LOAD ERROR LOOP ADDRESS
 2030  007170  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2031  007174  012706  001100          DSF4:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2032  007200  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2033  007204  052764  002000  000022          BIS     #BIT10,RPM3(R4) ;SET MAINT. SEEK INCOMPLETE
 2034  007212  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
```

```
2035  007216  032737  004000  001220          BIT     #BIT11,$RPDS    ;DID SEEK INCOMPLETE SET?
2036  007224  001002                          BNE     1$
2037  007226  104011                          ERROR   11              ;SEEK INCOMPLETE DID NOT SET
2038  007230  000426                          BR      4$              ;BYPASS REST OF THE TEST
2039  007232  032737  000001  001222  1$:     BIT     #BIT00,$RPER    ;DID DISK ERROR SET?
2040  007240  001001                          BNE     2$              ;BR IF YES
2041  007242  104012                          ERROR   12              ;DISK ERROR DID NOT SET AFTER SUSI
2042  007244                          2$:
2043  007244  013746  001224                  MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
2044  007250  005116                          COM     (SP)            ;COMPLEMENT THE CONTENTS
2045  007252  032726  140000                  BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'HE' BITS
2046  007256  001401                          BEQ     3$              ;BR IF BOTH SET
2047  007260  104013                          ERROR   13              ;'ERR' OR 'HE' DIDN'T SET WITH 'SUSI'
2048  007262  042764  002000  000022  3$:     BIC     #BIT10,RPM3(R4) ;CLEAR 'SUSI'
2049  007270  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2050  007274  032737  004000  001220          BIT     #BIT11,$RPDS    ;DID SEEK INCOMPLETE CLEAR?
2051  007302  001401                          BEQ     4$              ;BRANCH IF CLEAR
2052  007304  104045                          ERROR   45              ;SEEK INCOMPLETE DID NOT CLEAR
2053  007306  000004                  4$:     SCOPE                   ;LOOP ?
2054
2055                                  ;;**************************************************************
2056                                  ;*TEST 15        TEST SETTING OF 'SUSU' (SELECTED UNIT SEEK UNDERWAY
2057
2058                                  ;;**************************************************************
2059  007310                          TST15:
2060  007310  000240                          NOP
2061  007312  012737  007332  001106          MOV     #DSF6,$LPADR    ;LOAD LOOP ON TEST ADDRESS
2062  007320  012737  007332  001110          MOV     #DSF6,$LPERR    ;LOAD ERROR LOOP ADDRESS
2063  007326  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2064  007332  012706  001100          DSF6:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
2065  007336  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2066  007342  042764  040000  000022          BIC     #BIT14,RPM3(R4) ;RESET MAINT READY
2067  007350  052764  020000  000022          BIS     #BIT13,RPM3(R4) ;SET MAINT DRIVE ONLINE
2068  007356  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2069  007362  032737  002000  001220          BIT     #BIT10,$RPDS    ;DID 'SUSU' SET ?
2070  007370  001002                          BNE     1$
2071  007372  104104                          ERROR   104             ;'SUSU' DIDN'T SET
2072  007374  000412                          BR      2$              ;BYPASS REST OF THE TEST
2073  007376  052764  040000  000022  1$:     BIS     #BIT14,RPM3(R4) ;SET MAINT READY
2074  007404  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2075  007410  032737  002000  001220          BIT     #BIT10,$RPDS    ;DID 'SUSU' RESET ?
2076  007416  001401                          BEQ     2$              ;BR IF IT DID
2077  007420  104105                          ERROR   105             ;'SUSU' DIDN'T CLEAR
2078  007422  000004                  2$:     SCOPE                   ;LOOP ?
2079
2080                                  ;;**************************************************************
2081                                  ;*TEST 16        TEST SETTING OF 'SUFU' (SELECTED UNIT FILE UNSAFE
2082
2083                                  ;;**************************************************************
2084  007424                          TST16:
2085  007424  000240                          NOP
2086  007426  012737  007446  001106          MOV     #DSF7,$LPADR    ;LOAD LOOP ON TEST ADDRESS
2087  007434  012737  007446  001110          MOV     #DSF7,$LPERR    ;LOAD ERROR LOOP ADDRESS
2088  007442  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2089  007446  012706  001100          DSF7:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
2090  007452  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
```

```
2091  007456  052764  004000  000022          BIS     #BIT11,RPM3(R4) ;SET MAINT. FILE UNSAFE
2092  007464  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2093  007470  032737  001000  001220          BIT     #BIT09,$RPDS    ;DID FILE UNSAFE SET?
2094  007476  001002                          BNE     1$
2095  007500  104014                          ERROR   14              ;FILE UNSAFE DID NOT SET
2096  007502  000412                          BR      2$              ;BYPASS REST OF THE TEST
2097  007504  042764  004000  000022  1$:     BIC     #BIT11,RPM3(R4) ;CLEAR 'SUFU'
2098  007512  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2099  007516  032737  001000  001220          BIT     #BIT09,$RPDS    ;IS FILE UNSAFE CLEAR?
2100  007524  001401                          BEQ     2$
2101  007526  104015                          ERROR   15              ;FILE UNSAFE DID NOT CLEAR
2102  007530  000004                  2$:     SCOPE                   ;LOOP ?
2103
2104                                          ;;*****************************************************************
2105                                          ;*TEST 17       TEST 'SUWP' (SELECTED UNIT WRITE PROTECTED)
2106
2107                                          ;;*****************************************************************
2108  007532                                  TST17:
2109  007532  000240                          NOP
2110  007534  012737  007554  001106          MOV     #DSF11,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2111  007542  012737  007554  001110          MOV     #DSF11,$LPERR   ;LOAD ERROR LOOP ADDRESS
2112  007550  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2113  007554  012706  001100          DSF11:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2114  007560  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2115  007564  052764  100000  000022          BIS     #BIT15,RPM3(R4) ;SET MAINT READ ONLY
2116  007572  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2117  007576  032737  000400  001220          BIT     #BIT08,$RPDS    ;DID WRITE PROTECT SET?
2118  007604  001002                          BNE     1$
2119  007606  104016                          ERROR   16              ;WRITE PROTECT DID NOT SET
2120  007610  000412                          BR      2$              ;BYPASS REST OF THE TEST
2121  007612  042764  100000  000022  1$:     BIC     #BIT15,RPM3(R4) ;CLEAR MAINTENANCE READ ONLY
2122  007620  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2123  007624  032737  000400  001220          BIT     #BIT08,$RPDS    ;DID WRITE PROTECT CLEAR?
2124  007632  001401                          BEQ     2$
2125  007634  104017                          ERROR   17              ;WRITE PROTECT DID NOT CLEAR
2126  007636  000004                  2$:     SCOPE                   ;LOOP ?
```

```
2127
2128                                    ;;*****************************************************************
2129                                    ;*TEST 20       TEST SET AND CLEAR OF THE ATTENTION BITS
2130
2131                                    ;;*****************************************************************
2132  007640                            TST20:
2133  007640  000240                           NOP
2134  007642  012737  007670  001106           MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
2135  007650  012737  007702  001110           MOV     #DSF13,$LPERR   ;LOAD ERROR LOOP ADDRESS
2136  007656  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2137  007662  012737  000012  001162           MOV     #10.,$TIMES     ;DO 10. ITERATIONS
2138  007670  012706  001100           1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
2139  007674  012737  000001  001124           MOV     #1,$GDDAT       ;INITIALIZE ATTENTION BIT PATTERN
2140  007702  012737  007702  001110  DSF13:  MOV     #DSF13,$LPERR   ;CHANGE LOOP ON ERROR ADDRESS
2141  007710  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2142  007714  012764  000377  000000           MOV     #377,RPDS(R4)   ;CLEAR ATTENTION BITS
2143  007722  013764  001124  000020           MOV     $GDDAT,RPM2(R4) ;SET MAINT ATTENTION BIT
2144  007730  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2145  007734  123737  001124  001220           CMPB    $GDDAT,$RPDS    ;DID THE ATTN BIT SET IN RPDS?
2146  007742  001404                           BEQ     2$              ;BRANCH IF OK
2147  007744  113737  001220  001126           MOVB    $RPDS,$BDDAT
2148  007752  104020                           ERROR   20              ;ATTENTION BIT DID NOT SET
2149  007754  012737  007754  001110  2$:     MOV     #2$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2150  007762  013764  001124  000000           MOV     $GDDAT,RPDS(R4) ;CLEAR ATTENTION BIT
2151  007770  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2152  007774  105737  001220                  TSTB    $RPDS           ;DID IT CLEAR?
2153  010000  001401                           BEQ     3$              ;BRANCH IF CLEAR
2154  010002  104021                           ERROR   21              ;ATTENTION BIT DID NOT CLEAR
2155  010004  006337  001124           3$:     ASL     $GDDAT          ;ROTATE PATTERN
2156  010010  032737  000400  001124           BIT     #BIT08,$GDDAT   ;END OF PATTERN?
2157  010016  001731                           BEQ     DSF13           ;BRANCH IF NO
2158  010020  012737  010020  001110  4$:     MOV     #4$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2159  010026  012764  000377  000020           MOV     #377,RPM2(R4)   ;SET ATTENTION BITS
2160  010034  000005                           RESET
2161  010036  105737  001220                  TSTB    $RPDS           ;DID RESET CLEAR ATTN BITS?
2162  010042  001401                           BEQ     5$
2163  010044  104022                           ERROR   22              ;RESET DID NOT CLEAR ATTENTION BITS
2164  010046  005064  000020           5$:     CLR     RPM2(R4)
2165  010052  012764  000377  000020           MOV     #377,RPM2(R4)   ;SET ALL ATTENTION BITS
2166  010060  005064  000000                   CLR     RPDS(R4)        ;ISSUE CLEAR RPDS
2167  010064  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2168  010070  122737  000377  001220           CMPB    #377,$RPDS      ;DID ATTENTION BITS REMAIN SET?
2169  010076  001410                           BEQ     6$              ;BRANCH IF YES
2170  010100  005037  001124                  CLR     $GDDAT
2171  010104  005037  001126                  CLR     $BDDAT
2172  010110  113737  001220  001126           MOVB    $RPDS,$BDDAT    ;GET BAD DATA
2173  010116  104023                           ERROR   23              ;ATTENTION BITS CLEARED WITH A ZERO
2174  010120  000004                   6$:     SCOPE                   ;LOOP ?
2175
2176                                    ;;*****************************************************************
2177                                    ;*TEST 21       TEST THE 'WPV' BIT (WRITE PROTECT VIOLATION)
2178
2179                                    ;;*****************************************************************
2180  010122                            TST21:
2181  010122  000240                           NOP
2182  010124  012737  010144  001106           MOV     #ERF1,$LPADR    ;LOAD LOOP ON TEST ADDRESS
```

```
 2183  010132  012737  010144  001110          MOV     #ERF1,$LPERR    ;LOAD ERROR LOOP ADDRESS
 2184  010140  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2185  010144  012706  001100          ERF1:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2186  010150  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2187  010154  052764  100000  000022          BIS     #BIT15,RPM3(R4) ;SET MAINT READ ONLY
 2188  010162  012764  177777  000006          MOV     #-1,RPWC(R4)    ;SET WORD COUNT
 2189  010170  112764  000003  000004          MOVB    #3,RPCS(R4)     ;ISSUE A WRITE
 2190  010176  012737  160001  001206          MOV     #BIT15!BIT14!BIT13!BIT00,TPL
 2191  010204  004737  021756                  JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
 2192  010210  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2193  010214  032737  100000  001222          BIT     #BIT15,$RPER    ;DID WRITE VIOLATION SET?
 2194  010222  001001                          BNE     1$
 2195  010224  104024                          ERROR   24              ;WRITE PROTECTION VIOLATION DID NOT SET
 2196  010226                          1$:
 2197  010226  013746  001224                  MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
 2198  010232  005116                          COM     (SP)            ;COMPLEMENT THE CONTENTS
 2199  010234  032726  140000                  BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'HE' BITS
 2200  010240  001401                          BEQ     2$              ;BR IF BOTH SET
 2201  010242  104025                          ERROR   25              ;'ERR' OR 'HE' NOT SET WITH 'WPV'
 2202  010244  000004                  2$:     SCOPE                   ;LOOP ?
 2203
 2204                                  ;;*****************************************************************
 2205                                  ;*TEST 22       TEST THE 'FUV' BIT (FILE UNSAFE VIOLATION)
 2206
 2207                                  ;;*****************************************************************
 2208  010246                          TST22:
 2209  010246  000240                          NOP
 2210  010250  012737  010270  001106          MOV     #ERF2,$LPADR    ;LOAD LOOP ON TEST ADDRESS
 2211  010256  012737  010270  001110          MOV     #ERF2,$LPERR    ;LOAD ERROR LOOP ADDRESS
 2212  010264  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2213  010270  012706  001100          ERF2:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2214  010274  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2215  010300  012764  177777  000006          MOV     #-1,RPWC(R4)    ;LOAD THE WORD COUNT
 2216  010306  112764  000003  000004          MOVB    #3,RPCS(R4)     ;ISSUE WRITE
 2217  010314  012737  064001  001206          MOV     #BIT14!BIT13:BIT11!BIT00,TPL
 2218  010322  004737  021756                  JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
 2219  010326  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2220  010332  032737  040000  001222          BIT     #BIT14,$RPER    ;DID FILE UNSAFE VIOLATION SET?
 2221  010340  001001                          BNE     1$
 2222  010342  104026                          ERROR   26              ;FILE UNSAFE VIOLATION DID NOT SET
 2223  010344                          1$:
 2224  010344  013746  001224                  MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
 2225  010350  005116                          COM     (SP)            ;COMPLEMENT THE CONTENTS
 2226  010352  032726  140000                  BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'HE' BITS
 2227  010356  001401                          BEQ     2$              ;BR IF BOTH SET
 2228  010360  104027                          ERROR   27              ;'ERR' OR 'HE' DID NOT SET WITH 'FUV'
 2229  010362  000004                  2$:     SCOPE                   ;LOOP ?
 2230
 2231                                  ;;*****************************************************************
 2232                                  ;*TEST 23       TEST 'NXC' BIT (NON-EXISTENT CYLINDER) WITH VALID ADDRESSES
 2233
 2234                                  ;;*****************************************************************
 2235  010364                          TST23:
 2236  010364  000240                          NOP
 2237  010366  012737  010406  001106          MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
 2238  010374  012737  010416  001110          MOV     #ERF3,$LPERR    ;LOAD ERROR LOOP ADDRESS
```

```
2239  010402  013704  001210                    MOV     RPADR,R4        ;RP11E BUS ADDRESS
2240  010406  012706  001100          1$:       MOV     #STACK,SP       ;SETUP THE STACK POINTER
2241  010412  005037  001124                    CLR     $GDDAT          ;START AT ADDRESS 0
2242  010416  004737  021706          ERF3:     JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2243  010422  013764  001124  000012            MOV     $GDDAT,RPCA(R4) ;LOAD CYLINDER ADDR
2244  010430  012737  177777  001226            MOV     #-1,$RPWC       ;LOAD WORD COUNT
2245  010436  112764  000003  000004            MOVB    #3,RPCS(R4)     ;ISSUE A WRITE
2246  010444  012737  060001  001206            MOV     #BIT14!BIT13!BIT00,TPL
2247  010452  004737  021756                    JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
2248  010456  004737  022014                    JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2249  010462  032737  020000  001222            BIT     #BIT13,$RPER    ;DID NON-EXISTENT CYLINDER SET?
2250  010470  001401                            BEQ     2$              ;BR IF NOT
2251  010472  104030                            ERROR   30              ;'NXC' SET ON VALID ADDRRESS
2252  010474  005237  001124          2$:       INC     $GDDAT          ;UPDATE CYLINDER ADDRESS
2253  010500  023737  001176  001124            CMP     MAXCYL,$GDDAT   ;IS ADDR STILL LEGAL?
2254  010506  103343                            BHIS    ERF3            ;BRANCH IF YES
2255  010510  000004                            SCOPE                   ;LOOP ?
2256
2257                                    ;;**************************************************************
2258                                    ;*TEST 24       TEST 'NXC' WITH INVALID ADDRESSES
2259
2260                                    ;;**************************************************************
2261  010512                            TST24:
2262  010512  000240                            NOP
2263  010514  012737  010534  001106            MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
2264  010522  012737  010552  001110            MOV     #ERF4,$LPERR    ;LOAD ERROR LOOP ADDRESS
2265  010530  013704  001210                    MOV     RPADR,R4        ;RP11E BUS ADDRESS
2266  010534  012706  001100          1$:       MOV     #STACK,SP       ;SETUP THE STACK POINTER
2267  010540  013737  001176  001124            MOV     MAXCYL,$GDDAT   ;MAXIMUM CYLINDER ADDRESS
2268  010546  005237  001124                    INC     $GDDAT          ;INCREMENT BEYOND LIMIT
2269  010552  004737  021706          ERF4:     JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2270  010556  013764  001124  000012            MOV     $GDDAT,RPCA(R4) ;LOAD CYLINDER ADDR
2271  010564  012764  177777  000006            MOV     #-1,RPWC(R4)    ;LOAD WORD COUNT
2272  010572  112764  000003  000004            MOVB    #3,RPCS(R4)     ;ISSUE WRITE
2273  010600  012737  060001  001206            MOV     #BIT14!BIT13!BIT00,TPL
2274  010606  004737  021756                    JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
2275  010612  004737  022014                    JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2276  010616  032737  020000  001222            BIT     #BIT13,$RPER    ;DID NON-EXISTENT CYL SET?
2277  010624  001002                            BNE     2$
2278  010626  104031                            ERROR   31              ;'NXC' DID NOT SET
2279  010630  000407                            BR      3$              ;BYPASS REST OF THE TEST
2280  010632                          2$:
2281  010632  013746  001224                    MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
2282  010636  005116                            COM     (SP)            ;COMPLEMENT THE CONTENTS
2283  010640  032726  140000                    BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'NE' BITS
2284  010644  001401                            BEQ     3$              ;BR IF BOTH SET
2285  010646  104032                            ERROR   32              ;'ERR' OR 'NE' DIDN'T SET WITH 'NXC'
2286  010650  005237  001124          3$:       INC     $GDDAT          ;UPDATE CYLINDER ADDR
2287  010654  023737  001200  001124            CMP     MAXPAT,$GDDAT   ;PATTERN EXCEEDED?
2288  010662  001333                            BNE     ERF4            ;BR IF NOT
2289  010664  000004                            SCOPE                   ;LOOP ?
2290
2291                                    ;;**************************************************************
2292                                    ;*TEST 25       TEST 'NXT' BIT (NON-EXISTENT TRACK) WITH VALID ADDRESSES
2293
2294                                    ;;**************************************************************
```

```
 2295  010666                                  TST25:
 2296  010666  000240                                  NOP
 2297  010670  012737  010710  001106                  MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
 2298  010676  012737  010720  001110                  MOV     #ERF5,$LPERR    ;LOAD ERROR LOOP ADDRESS
 2299  010704  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2300  010710  012706  001100          1$:             MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2301  010714  005037  001124                          CLR     $GDDAT          ;STARTING TRACK ADDR OF 0
 2302  010720  004737  021706          ERF5:           JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2303  010724  113764  001124  000015                  MOVB    $GDDAT,RPDA+1(R4)  ;LOAD TRACK ADDRESS
 2304  010732  012764  177777  000006                  MOV     #-1,RPWC(R4)    ;LOAD WORD COUNT
 2305  010740  112764  000003  000004                  MOVB    #3,RPCS(R4)     ;WRITE
 2306  010746  012737  060001  001206                  MOV     #BIT14!BIT13!BIT00,TPL
 2307  010754  004737  021756                          JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
 2308  010760  004737  022014                          JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2309  010764  032737  010000  001222                  BIT     #BIT12,$RPER    ;IS NXT SET?
 2310  010772  001401                                  BEQ     2$
 2311  010774  104033                                  ERROR   33              ;'NXT' SET ON VALID ADDRESS
 2312  010776  005237  001124          2$:             INC     $GDDAT          ;INCREMENT THE TRACK ADDRESS
 2313  011002  022737  000023  001124                  CMP     #19.,$GDDAT     ;IS TRACK ADDRESS STILL VALID
 2314  011010  103343                                  BHIS    ERF5            ;BRANCH IF YES
 2315  011012  000004                                  SCOPE                   ;LOOP ?
 2316
 2317
 2318                                          ;;**************************************************************
 2319                                          ;*TEST 26        TEST 'NXT' BIT WITH INVALID ADDRESSES
 2320
 2321                                          ;;**************************************************************
 2322  011014                                  TST26:
 2323  011014  000240                                  NOP
 2324  011016  012737  011036  001106                  MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
 2325  011024  012737  011050  001110                  MOV     #ERF6,$LPERR    ;LOAD ERROR LOOP ADDRESS
 2326  011032  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2327  011036  012706  001100          1$:             MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2328  011042  012737  000024  001124                  MOV     #20.,$GDDAT     ;START WITH INVALID ADDRESS
 2329  011050  004737  021706          ERF6:           JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2330  011054  113764  001124  000015                  MOVB    $GDDAT,RPDA+1(R4)  ;LOAD TRACK ADDR
 2331  011062  012737  177777  001226                  MOV     #-1,$RPWC
 2332  011070  012764  000003  000004                  MOV     #3,RPCS(R4)     ;WRITE
 2333  011076  012737  060001  001206                  MOV     #BIT14!BIT13!BIT00,TPL
 2334  011104  004737  021756                          JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
 2335  011110  004737  022014                          JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2336  011114  032737  010000  001222                  BIT     #BIT12,$RPER    ;DID NXT SET?
 2337  011122  001002                                  BNE     2$
 2338  011124  104034                                  ERROR   34              ;'NXT' DIDN'T SET WITH INVALID ADDRESS
 2339  011126  000407                                  BR      3$              ;BYPASS REST OF THE TEST
 2340  011130                                  2$:
 2341  011130  013746  001224                          MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
 2342  011134  005116                                  COM     (SP)            ;COMPLEMENT THE CONTENTS
 2343  011136  032726  140000                          BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'HE' BITS
 2344  011142  001401                                  BEQ     3$              ;BR IF BOTH SET
 2345  011144  104035                                  ERROR   35              ;'ERR' OR 'HE' DIDN'T SET WITH 'NXT'
 2346  011146  005237  001124          3$:             INC     $GDDAT          ;INCREMENT TRACK ADDR.
 2347  011152  022737  000040  001124                  CMP     #40,$GDDAT      ;END OF PATTERN?
 2348  011160  001333                                  BNE     ERF6            ;BRANCH IF NOT
 2349  011162  000004                                  SCOPE                   ;LOOP ?
 2350
```

```
                                        ;;*****************************************************************
                                        ;*TEST 27       TEST 'NXS' BIT (NON-EXISTENT SECTOR) WITH VALID ADDRESSES
                                        ;
                                        ;;*****************************************************************
011164                                  TST27:
011164  000240                                NOP
011166  012737  011206  001106                MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
011174  012737  011216  001110                MOV     #ERF7,$LPERR    ;LOAD ERROR LOOP ADDRESS
011202  013704  001210                        MOV     RPADR,R4        ;RP11E BUS ADDRESS
011206  012706  001100          1$:           MOV     #STACK,SP       ;SETUP THE STACK POINTER
011212  005037  001124                        CLR     $GDDAT          ;STARTING SECTOR ADDRESS OF 0
011216  004737  021706          ERF7:         JSR     PC,CLRP         ;CLEAR THE CONTROLLER
011222  013764  001124  000014                MOV     $GDDAT,RPDA(R4) ;LOAD SECTOR ADDR
011230  012764  177777  000006                MOV     #-1,RPWC(R4)    ;LOAD WORD COUNT
011236  112764  000003  000004                MOVB    #3,RPCS(R4)     ;WRITE
011244  012737  060001  001206                MOV     #BIT14!BIT13!BIT00,TPL
011252  004737  021756                        JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
011256  004737  022014                        JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
011262  032737  004000  001222                BIT     #BIT11,$RPER    ;DID NXS SET?
011270  001401                                BEQ     2$
011272  104036                                ERROR   36              ;'NXS' SET ON VALID ADDRESS
011274  005237  001124          2$:           INC     $GDDAT          ;UPDATE SECTOR ADDR
011300  022737  000011  001124                CMP     #9.,$GDDAT      ;IS ADDR STILL LEGAL?
011306  103343                                BHIS    ERF7            ;BRANCH IF YES
011310  000004                                SCOPE                   ;LOOP ?

                                        ;;*****************************************************************
                                        ;*TEST 30       TEST 'NXS' BIT WITH INVALID ADDRESSES
                                        ;
                                        ;;*****************************************************************
011312                                  TST30:
011312  000240                                NOP
011314  012737  011334  001106                MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
011322  012737  011346  001110                MOV     #ERF10,$LPERR   ;LOAD ERROR LOOP ADDRESS
011330  013704  001210                        MOV     RPADR,R4        ;RP11E BUS ADDRESS
011334  012706  001100          1$:           MOV     #STACK,SP       ;SETUP THE STACK POINTER
011340  012737  000012  001124                MOV     #10.,$GDDAT     ;START WITH MAXIMUM SECTOR ADDR + 1
011346  004737  021706          ERF10:        JSR     PC,CLRP         ;CLEAR THE CONTROLLER
011352  013764  001124  000014                MOV     $GDDAT,RPDA(R4) ;LOAD SECTOR ADDR
011360  012764  177777  000006                MOV     #-1,RPWC(R4)    ;LOAD WORD COUNT
011366  112764  000003  000004                MOVB    #3,RPCS(R4)     ;WRITE
011374  012737  060001  001206                MOV     #BIT14!BIT13!BIT00,TPL
011402  004737  021756                        JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
011406  004737  022014                        JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
011412  032737  004000  001222                BIT     #BIT11,$RPER    ;DID NXS SET?
011420  001002                                BNE     2$
011422  104037                                ERROR   37              ;'NXS' DIDN'T SET WITH INVALID ADDRESS
011424  000407                                BR      3$              ;BYPASS REST OF THE TEST
011426                          2$:
011426  013746  001224                        MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
011432  005116                                COM     (SP)            ;COMPLEMENT THE CONTENTS
011434  032726  140000                        BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'HE' BITS
011440  001401                                BEQ     3$              ;BR IF BOTH SET
011442  104040                                ERROR   40              ;'ERR' OR 'HE' DIDN'T SET WITH 'NXS'
011444  005237  001124          3$:           INC     $GDDAT          ;UPDATE ADDRESS
011450  022737  000020  001124                CMP     #20,$GDDAT      ;IS PATTERN EXHAUSTED?
```

```
2407  011456  001333                          BNE     ERF10           ;BRANCH IF NOT
2408  011460  000004                          SCOPE                   ;LOOP ?
2409
2410                                  ;;*****************************************************************
2411                                  ;*TEST 31       TEST SETTING OF 'PROG' BIT (PROGRAM ERROR)
2412
2413                                  ;;*****************************************************************
2414  011462                          TST31:
2415  011462  000240                          NOP
2416  011464  012737  011504  001106          MOV     #ERF11,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2417  011472  012737  011504  001110          MOV     #ERF11,$LPERR   ;LOAD ERROR LOOP ADDRESS
2418  011500  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2419  011504  012706  001100          ERF11:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2420  011510  004737  021706                  JSR     PC,CLRP         ;CLEAR RP11
2421  011514  042764  020000  000022          BIC     #BIT13,RPM3(R4) ;RESET MAINTENANCE 'SUOL'
2422  011522  012764  000003  000004          MOV     #3,RPCS(R4)     ;ISSUE WRITE
2423  011530  012737  060001  001206          MOV     #BIT14!BIT13!BIT0,TPL
2424  011536  004737  021756                  JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
2425  011542  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2426  011546  032737  002000  001222          BIT     #BIT10,$RPER    ;DID PROGRAM ERROR SET?
2427  011554  001001                          BNE     1$
2428  011556  104041                          ERROR   41              ;'PROG' DIDN'T SET
2429  011560                          1$:
2430  011560  013746  001224                  MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
2431  011564  005116                          COM     (SP)            ;COMPLEMENT THE CONTENTS
2432  011566  032726  140000                  BIT     #BIT15!BIT14,(SP)+  ;CHECK 'ERR' AND 'HE' BITS
2433  011572  001401                          BEQ     2$              ;BR IF BOTH SET
2434  011574  104042                          ERROR   42              ;'ERR' OR 'HE' DIDN'T SET WITH 'PROG'
2435  011576  032764  020000  000022  2$:     BIT     #BIT13,RPM3(R4) ;SET MAINTENANCE 'SUOL'
2436  011604  000004                          SCOPE                   ;LOOP ?
2437
2438
2439                                  ;;*****************************************************************
2440                                  ;*TEST 32       TEST SETTING OF 'MODE' BIT (MODE ERROR)
2441
2442                                  ;;*****************************************************************
2443  011606                          TST32:
2444  011606  000240                          NOP
2445  011610  012737  011630  001106          MOV     #ERF12,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2446  011616  012737  011630  001110          MOV     #ERF12,$LPERR   ;LOAD ERROR LOOP ADDRESS
2447  011624  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2448  011630  012706  001100          ERF12:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2449  011634  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2450  011640  012764  177777  000006          MOV     #-1,RPWC(R4)    ;LOAD WORD COUNT
2451  011646  012764  004003  000004          MOV     #4003,RPCS(R4)  ;WRITE HEADER IN PDP11 MODE
2452  011654  012737  060001  001206          MOV     #BIT14!BIT13!BIT00,TPL
2453  011662  004737  021756                  JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
2454  011666  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2455  011672  032737  000400  001222          BIT     #BIT08,$RPER    ;DID MODE ERROR SET?
2456  011700  001001                          BNE     1$
2457  011702  104043                          ERROR   43              ;'MODE' ERROR DIDN'T SET
2458  011704                          1$:
2459  011704  013746  001224                  MOV     $RPCS,-(SP)     ;PUT CONTENTS OF RPCS ON THE STACK
2460  011710  005116                          COM     (SP)            ;COMPLEMENT THE CONTENTS
2461  011712  032726  140000                  BIT     #BIT15!BIT14,(SP)+  ;CHECK ERR' AND 'HE' BITS
2462  011716  001401                          BEQ     2$              ;BR IF BOTH SET
```

```
                                        ERROR   44              ;'ERR' OR 'HE' DIDN'T SET WITH 'MODE'
                                2$:     SCOPE                   ;LOOP '

                                ;;**************************************************************
                                ;*TEST 33        TEST ATTENTION INTERRUPT

                                ;;**************************************************************
                                TST33:
                                        NOP
                                        MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
                                        MOV     #CSF1,$LPERR    ;LOAD ERROR LOOP ADDRESS
                                        MOV     RPADR,R4        ;RP11E BUS ADDRESS
                                1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
                                        MOV     #1,$GDOAT       ;STARTING TEST PATTERN
                                CSF1:   JSR     PC,CLRP         ;CLEAR THE CONTROLLER
                                        MOV     #PR7,@#PVEC+2
                                        MOV     #2$,@#PVEC      ;INTERRUPT RETURN ADDRESS
                                        JSR     PC,CLRP         ;CLEAR RP11
                                        CLR     @#PSW           ;CLEAR PRIORITY LEVEL
                                        BIS     #BIT13,RPCS(R4) ;ENABLE ATTENTION INTERRUPT
                                        MOV     $GDOAT,RPM2(R4) ;SET ATTENTION BIT AND
                                        NOP                     ;WAIT FOR INTERRUPT
                                        NOP
                                        NOP
                                        MOV     #PR7,@#PSW      ;LOCKOUT INTERRUPTS
                                        JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
                                        ERROR   46              ;NO ATTENTION INTERRUPT
                                        BR      3$              ;BYPASS REST OF THE TEST
                                2$:     MOV     #STACK,SP       ;RESTORE STACK
                                        BIT     #BIT13,RPCS(R4) ;DID 'AIE' CLEAR?
                                        BEQ     3$
                                        ERROR   47              ;'AIE' DIDN'T CLEAR WHEN INTERRUPT OCCURED
                                3$:     MOV     $GDOAT,RPDS(R4) ;CLEAR ATTENTION BIT
                                        ASL     $GDOAT          ;SHIFT TEST PATTERN
                                        BIT     #BIT08,$GDOAT   ;PATTERN EXCEEDED?
                                        BEQ     CSF1            ;BRANCH IF NO
                                        SCOPE                   ;LOOP '

                                ;;**************************************************************
                                ;*TEST 34        TEST NO ATTENTION INTERRUPT

                                ;;**************************************************************
                                TST34:
                                        NOP
                                        MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
                                        MOV     #TSTNAT,$LPERR  ;LOAD ERROR LOOP ADDRESS
                                        MOV     RPADR,R4        ;RP11E BUS ADDRESS
                                1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
                                        MOV     #1,$GDOAT       ;STARTING TEST PATTERN
                                TSTNAT: JSR     PC,CLRP         ;CLEAR THE CONTROLLER
                                        MOV     #PR7,@#PVEC+2
                                        MOV     #1$,@#PVEC      ;INTERRUPT RETURN ADDRESS
                                        JSR     PC,CLRP         ;CLEAR RP11
                                        BIS     #BIT13,RPCS(R4) ;SET 'AIE'
                                        CLR     @#PSW           ;CLEAR PRIORITY LEVEL
                                        NOP                     ;WAIT FOR INTERRUPT
```

```
2519  012206  000240                          NOP
2520  012210  000240                          NOP
2521  012212  012737  000340  177776          MOV     #PR7,@#PSW      ;LOCKOUT INTERRUPTS
2522  012220  000405                          BR      2$              ;OK, INTERRUPT DIDN'T OCCUR
2523  012222  012706  001100          1$:     MOV     #STACK,SP       ;RESTORE STACK
2524  012226  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2525  012232  104107                          ERROR   107             ;ATTENTION INTERRUPT OCCURED WITH NO ATTN BITS SET
2526  012234  000004                  2$:     SCOPE                   ;LOOP ?
2527
2528                                          ;;*****************************************************************
2529                                          ;*TEST 35       TEST ATTENTION INTERRUPT WITH 2 ATTN BITS SET
2530
2531                                          ;;*****************************************************************
2532  012236                                  TST35:
2533  012236  000240                          NOP
2534  012240  012737  012260  001106          MOV     #TSTAT,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2535  012246  012737  012260  001110          MOV     #TSTAT,$LPERR   ;LOAD ERROR LOOP ADDRESS
2536  012254  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2537  012260  012706  001100          TSTAT:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2540  012272  012777  000340  166714          MOV     #PR7,@RPVEC+2
2541  012300  012777  012356  166704          MOV     #1$,@RPVEC
2542  012306  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2543  012312  005037  177776                  CLR     @#PSW           ;LOWER PROCESSOR PRIORITY
2544  012316  052764  020000  000004          BIS     #BIT13,RPCS(R4) ;ENABLE ATTENTION INTERRUPT
2545  012324  012764  000003  000020          MOV     #3,RPM2(R4)     ;SET ATTENTION BITS
2546  012332  000240                          NOP
2547  012334  000240                          NOP
2548  012336  000240                          NOP
2549  012340  012737  000340  177776          MOV     #PR7,@#PSW      ;RAISE PROCESSOR PRIORITY
2550  012346  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2551  012352  104050                          ERROR   50              ;RP11E DID NOT INTERRUPT WITH ATTENTION
2552                                                                  ;BITS 0 AND 1 SET
2553  012354  000431                          BR      2$
2554  012356  012737  012356  001110  1$:     MOV     #1$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
2555  012364  012706  001100                  MOV     #STACK,SP       ;RESTORE STACK
2556  012370  012777  012440  166614          MOV     #2$,@RPVEC
2557  012376  005037  177776                  CLR     @#PSW
2560  012402  052764  020000  000004          BIS     #BIT13,RPCS(R4) ;ENABLE ATTENTION INTERRUPT
2561  012410  012764  000001  000000          MOV     #BIT00,RPDS(R4) ;CLEAR ATTENTION BIT ZERO
2562  012416  000240                          NOP
2563  012420  000240                          NOP
2564  012422  000240                          NOP
2565  012424  012737  000340  177776          MOV     #PR7,@#PSW      ;RAISE PROCESSOR PRIORITY
2566  012432  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2567  012436  104051                          ERROR   51              ;ATTENTION BIT 1 DID NOT INTERRUPT
2570  012440  000004                  2$:     SCOPE                   ;LOOP ?
2571
2572                                          ;;*****************************************************************
2573                                          ;*TEST 36       TEST 'IDE' BIT (INTERRUPT ON DONE ENABLE)
2574
2575                                          ;;*****************************************************************
2576  012442                                  TST36:
2577  012442  000240                          NOP
2600  012444  012737  012464  001106          MOV     #CSR2,$LPADR    ;LOAD LOOP ON TEST ADDRESS
```

```
012452 012737 012464 001110          MOV     #CSF2,$LPERR    ;LOAD ERROR LOOP ADDRESS
012460 013704 001210                 MOV     RP%OR,R4        ;RP11E BUS ADDRESS
012464 012706 001100          CSF2:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
012470 004737 021706                 JSR     PC,CLR%         ;CLEAR THE CONTROLLER
012474 012777 012546 166510          MOV     #1$,@RPVEC      ;RETURN VECTOR
012502 012777 000340 166504          MOV     #PR7,@RPVEC+2
012510 005037 177776                 CLR     PSW             ;ALLOW INTERRUPTS
012514 052764 000100 000004          BIS     #BIT06,RPCS(R4) ;ENABLE INTERRUPT ON READY
012522 000240                        NOP
012524 000240                        NOP
012526 000240                        NOP
012530 012737 000340 177776          MOV     #PR7,@#PSW      ;LOCKOUT INTERRUPTS
012536 004737 022014                 JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
012542 104052                        ERROR   52              ;NO READY INTERRUPT
012544 000407                        BR      2$              ;BYPASS 'IDE' CHECK
012546 004737 022014          1$:    JSR     PC,SAVRP        ;SAVE THE REGISTERS
012552 032737 000100 001224          BIT     #BIT06,$RPCS    ;IS 'IDE' STILL SET ?
012560 001001                        BNE     2$              ;BR IF IT IS
012562 104106                        ERROR   106             ;'IDE' NOT SET AFTER INTERRUPT
012564 000004                 2$:    SCOPE                   ;LOOP ?

                              ;;*****************************************************************
                              ;*TEST 37       TEST INTERRUPT WITHOUT INTERRUPT ENABLE SET

                              ;;*****************************************************************
012566                        TST37:
012566 000240                        NOP
012570 012737 012610 001106          MOV     #CSF3,$LPROR    ;LOAD LOOP ON TEST ADDRESS
012576 012737 012610 001110          MOV     #CSF3,$LPERR    ;LOAD ERROR LOOP ADDRESS
012604 013704 001210                 MOV     RP%OR,R4        ;RP11E BUS ADDRESS
012610 012706 001100          CSF3:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
012614 004737 021706                 JSR     PC,CLR%         ;CLEAR THE CONTROLLER
012620 012777 012656 166364          MOV     #1$,@RPVEC      ;INTERRUPT VECTOR
012626 112764 000377 000020          MOVB    #377,RPM2(R4)   ;SET ATTENTION BITS
012634 005037 177776                 CLR     @#PSW           ;ALLOW INTERRUPTS AND
012640 000240                        NOP                     ;WAIT AWHILE
012642 000240                        NOP
012644 000240                        NOP
012646 012737 000340 177776          MOV     #PR7,@#PSW      ;LOCKOUT INTERRUPTS
012654 000403                        BR      2$
012656 004737 022014          1$:    JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
012662 104053                        ERROR   53              ;READY INTERRUPT OCCURED WITHOUT 'IDE' SET
012664 000004                 2$:    SCOPE                   ;LOOP ?

                              ;;*****************************************************************
                              ;*TEST 40       TEST INTERRUPT AT PRIORITY 4

                              ;;*****************************************************************
012666                        TST40:
012666 000240                        NOP
012670 012737 012710 001106          MOV     #CSF6,$LPROR    ;LOAD LOOP ON TEST ADDRESS
012676 012737 012710 001110          MOV     #CSF6,$LPERR    ;LOAD ERROR LOOP ADDRESS
012704 013704 001210                 MOV     RP%OR,R4        ;RP11E BUS ADDRESS
012710 012706 001100          CSF6:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
012714 004737 021706                 JSR     PC,CLR%         ;CLEAR RP11E
012720 012777 012764 166264          MOV     #1$,@RPVEC      ;SETUP INTERRUPT VECTOR
```

```
2631  012726  052764  000100  000004          BIS     #BIT06,RPCS(R4) ;ENABLE READY INTERRUPT
2632  012734  012737  000200  177776          MOV     #PR4,@#PSW      ;LOWER PROCESSOR LEVEL
2633  012742  000240                          NOP
2634  012744  000240                          NOP
2635  012746  000240                          NOP
2636  012750  012737  000340  177776          MOV     #PR7,@#PSW
2637  012756  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2638  012762  104054                          ERROR   54              ;NO READY INTERRUPT AT LEVEL 4
2639  012764  000004                  1$:     SCOPE                   ;LOOP ?
2640
2641                                          ;;**************************************************************
2642                                          ;*TEST 41        TEST NO INTERRUPT AT PRIORITY 5
2643
2644                                          ;;**************************************************************
2645  012766                                  TST41:
2646  012766  000240                          NOP
2647  012770  012737  013010  001106          MOV     #CSF7,$LPADR    ;LOAD LOOP ON TEST ADDRESS
2648  012776  012737  013010  001110          MOV     #CSF7,$LPERR    ;LOAD ERROR LOOP ADDRESS
2649  013004  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2650  013010  012706  001100          CSF7:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
2651  013014  004737  021706                  JSR     PC,CLRP         ;CLEAR RP11E
2652  013020  012777  013060  166164          MOV     #1$,@RPVEC      ;SETUP INTERRUPT VECTOR
2653  013026  052764  000100  000004          BIS     #BIT06,RPCS(R4) ;ENABLE INTERRUPT ON READY
2654  013034  012737  000240  177776          MOV     #PR5,@#PSW      ;SET PRIORITY LEVEL TO 5
2655  013042  000240                          NOP
2656  013044  000240                          NOP
2657  013046  000240                          NOP
2658  013050  012737  000340  177776          MOV     #PR7,@#PSW
2659  013056  000403                          BR      2$
2660  013060  004737  022014          1$:     JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2661  013064  104055                          ERROR   55              ;INTERRUPT RECEIVED AT PRIORITY LEVEL 5
2662  013066  000004                  2$:     SCOPE                   ;LOOP ?
2663
2664                                          ;;**************************************************************
2665                                          ;*TEST 42        TEST NO INTERRUPT AT PRIORITY 6
2666
2667                                          ;;**************************************************************
2668  013070                                  TST42:
2669  013070  000240                          NOP
2670  013072  012737  013112  001106          MOV     #CSF10,$LPADR   ;LOAD LOOP ON TEST ADDRESS
2671  013100  012737  013112  001110          MOV     #CSF10,$LPERR   ;LOAD ERROR LOOP ADDRESS
2672  013106  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2673  013112  012706  001100          CSF10:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2674  013116  004737  021706                  JSR     PC,CLRP         ;CLEAR RP11E
2675  013122  012777  013162  166062          MOV     #1$,@RPVEC      ;TRAP VECTOR
2676  013130  052764  000100  000004          BIS     #BIT06,RPCS(R4) ;ENABLE READY INTERRUPT
2677  013136  012737  000300  177776          MOV     #PR6,@#PSW      ;SET PRIORITY LEVEL TO 6
2678  013144  000240                          NOP
2679  013146  000240                          NOP
2680  013150  000240                          NOP
2681  013152  012737  000340  177776          MOV     #PR7,@#PSW
2682  013160  000403                          BR      2$
2683  013162  004737  022014          1$:     JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2684  013166  104056                          ERROR   56              ;INTERRUPT RECEIVED AT LEVEL 6
2685  013170  000004                  2$:     SCOPE                   ;LOOP ?
2686
```

```
                                        ;;*****************************************************************
                                        ;*TEST 43        TEST NO INTERRUPT AT PRIORITY 7
                                        ;
                                        ;;*****************************************************************
013172                                  TST43:
013172  000240                                 NOP
013174  012737  013214  001106                 MOV     #CSF11,SLPADR   ;LOAD LOOP ON TEST ADDRESS
013202  012737  013214  001110                 MOV     #CSF11,SLPERR   ;LOAD ERROR LOOP ADDRESS
013210  013704  001210                         MOV     RPADR,R4        ;RP11E BUS ADDRESS
013214  012706  001100          CSF11:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
013220  004737  021706                         JSR     PC,CLRP         ;CLEAR RP11E
013224  012777  013250  165760                 MOV     #1$,@RPVEC      ;SETUP VECTOR INTERRUPT
013232  052764  000100  000004                 BIS     #BIT06,RPCS(R4) ;ENABLE READY INTERRUPT
013240  000240                                 NOP
013242  000240                                 NOP
013244  000240                                 NOP
013246  000403                                 BR      2$
013250  004737  022014          1$:     JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
013254  104056                                 ERROR   56              ;INTERRUPT RECEIVED AT LEVEL 7
013256  000004                          2$:     SCOPE                   ;LOOP ?
                                        ;;*****************************************************************
                                        ;*TEST 44        TEST CLEAR AND SET OF 'RDY' (CONTROLLER READY)
                                        ;
                                        ;;*****************************************************************
013260                                  TST44:
013260  000240                                 NOP
013262  012737  013302  001106                 MOV     #CSF12,SLPADR   ;LOAD LOOP ON TEST ADDRESS
013270  012737  013302  001110                 MOV     #CSF12,SLPERR   ;LOAD ERROR LOOP ADDRESS
013276  013704  001210                         MOV     RPADR,R4        ;RP11E BUS ADDRESS
013302  012706  001100          CSF12:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
013306  004737  021706                         JSR     PC,CLRP         ;CLEAR RP11E
013312  112764  000003  000004                 MOVB    #3,RPCS(R4)     ;ISSUE WRITE
013320  004737  022014                         JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
013324  105737  001224                         TSTB    @RPCS           ;IS READY SET?
013330  100001                                 BPL     1$              ;BRANCH IF NO
013332  104101                                 ERROR   101             ;'READY' DID NOT CLEAR WITH 'GO'
013334  012737  060001  001206  1$:     MOV     #BIT14!BIT13!BIT00,TPL
013342  004737  021756                         JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
013346  004737  022014                         JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
013352  105737  001224                         TSTB    @RPCS           ;DID READY SET?
013356  100401                                 BMI     2$
013360  104060                                 ERROR   60              ;READY DID NOT SET AT END OF OPERATION
013362  000004                          2$:     SCOPE                   ;LOOP ?
                                        ;;*****************************************************************
                                        ;*TEST 45        TEST 'SUCA' BIT PATH IN THE CONTROLLER
                                        ;
                                        ;;*****************************************************************
013364                                  TST45:
013364  000240                                 NOP
013366  012737  013406  001106                 MOV     #1$,SLPADR      ;LOAD LOOP ON TEST ADDRESS
013374  012737  013416  001110                 MOV     #CAF1,SLPERR    ;LOAD ERROR LOOP ADDRESS
013402  013704  001210                         MOV     RPADR,R4        ;RP11E BUS ADDRESS
013406  012706  001100          1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
013412  005037  001124                         CLR     @GDDAT          ;START TEST PATTERN AT 0
```

```
2743  013416  004737  021706          CAF1:   JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2744  013422  113764  001124  000021          MOVB    $GDDAT,RPM2+1(R4)  ;LOAD MAINT CYLINDER ADDR
2745  013430  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2746  013434  013737  001240  001126          MOV     $SUCA,$GDDAT    ;GET DISK CYLINDER ADDR
2747  013442  023737  001124  001126          CMP     $GDDAT,$GDDAT   ;IS SUCA CORRECT?
2748  013450  001401                          BEQ     2$
2749  013452  104061                          ERROR   61              ;SUCA INCORRECT
2750  013454  005237  001124          2$:     INC     $GDDAT          ;UPDATE CYLINDER ADDR.
2751  013460  032737  000400  001124          BIT     #BIT08,$GDDAT   ;IS PATTERN EXCEEDED?
2752  013466  001753                          BEQ     CAF1            ;BRANCH IF NO
2753  013470  000004                          SCOPE                   ;LOOP ?
2754
2755                                  ;;**************************************************************
2756                                  ;*TEST 46        TEST THAT THE 'SOT' COUNTS CORRECTLY
2757
2758                                  ;;**************************************************************
2759  013472                          TST46:
2760  013472  000240                          NOP
2761  013474  012737  013530  001106          MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
2762  013502  012737  013560  001110          MOV     #DAF1,$LPERR    ;LOAD ERROR LOOP ADDRESS
2763  013510  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2764  013514  012764  020400  000022          MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE TO CLEAR
2765  013522  012764  020000  000022          MOV     #BIT13,RPM3(R4) ;INDEX SYNC FF (IF SET)
2766  013530  012706  001100          1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
2767  013534  016437  000014  001124          MOV     RPDA(R4),$GDDAT  ;GET THE REGISTER
2768  013542  042737  177417  001124          BIC     #^C360,$GDDAT   ;LEAVE THE 'SOT' BITS
2769  013550  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2770  013554  012705  000024                  MOV     #20.,R5         ;ITERATION COUNTER
2771  013560  004737  022014          DAF1:   JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2772  013564  013737  001234  001126          MOV     $RPDA,$GDDAT    ;GET SOT
2773  013572  013737  001234  001160          MOV     $RPDA,$TMP0     ;MOVE CONTENTS TO A WORKING LOCATION
2774  013600  042737  177417  001160          BIC     #^C360,$TMP0    ;CLEAR UNWANTED BITS
2775  013606  023737  001124  001160          CMP     $GDDAT,$TMP0    ;CONTENTS OF 'SOT' CORRECT ?
2776  013614  001401                          BEQ     2$
2777  013616  104063                          ERROR   63              ;CONTENTS OF SOT INCORRECT
2778  013620  005305                  2$:     DEC     R5              ;DECREMENT THE ITERATION COUNTER
2779  013622  001420                          BEQ     3$              ;BR WHEN FINISHED
2780  013624  012764  020400  000022          MOV     #BIT13!BIT08,RPM3(R4)   ;GENERATE ONE SECTOR PULSE
2781  013632  012764  020000  000022          MOV     #BIT13,RPM3(R4)
2782  013640  062737  000020  001124          ADD     #20,$GDDAT      ;UPDATE TEST ADDR
2783  013646  022737  000360  001124          CMP     #360,$GDDAT     ;MAXIMUM VALUE ?
2784  013654  103341                          BHIS    DAF1            ;BRANCH IF NOT
2785  013656  005037  001124                  CLR     $GDDAT          ;RESET TO ZERO
2786  013662  000736                          BR      DAF1            ;CONTINUE
2787  013664  000004                  3$:     SCOPE                   ;LOOP ?
2788
2789                                  ;;**************************************************************
2790                                  ;*TEST 47        TEST THAT 'INDEX' CLEARS THE 'SOT' COUNTER
2791
2792                                  ;;**************************************************************
2793  013666                          TST47:
2794  013666  000240                          NOP
2795  013670  012737  013710  001106          MOV     #DAF2,$LPADR    ;LOAD LOOP ON TEST ADDRESS
2796  013676  012737  013710  001110          MOV     #DAF2,$LPERR    ;LOAD ERROR LOOP ADDRESS
2797  013704  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
2798  013710  012706  001100          DAF2:   MOV     #STACK,SP       ;SETUP THE STACK POINTER
```

```
 2799  013714  004737  021706                JSR     PC,CLRP         ;CLEAR RP11E
 2800  013720  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2801  013726  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2802  013734  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2803  013742  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2804  013750  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2805  013756  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2806  013764  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2807  013772  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2808  014000  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2809  014006  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2810  014014  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2811  014022  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2812  014030  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2813  014036  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2814  014044  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2815  014052  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2816  014060  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2817  014066  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2818  014074  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2819  014102  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2820  014110  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2821  014116  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2822  014124  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2823  014132  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2824  014140  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2825  014146  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2826  014154  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2827  014162  012764  020000  000022        MOV     #BIT13,RPM3(R4)
 2828  014170  004037  021766                JSR     R0,INDEXP       ;GENERATE 1 INDEX PULSE
 2829  014174  000001                        .WORD   1               ;CONSTANT FOR 1 INDEX PULSE
 2830  014176  012764  020400  000022        MOV     #BIT13!BIT08,RPM3(R4)  ;GENERATE A SECTOR PULSE
 2831  014204  012764  020000  000022        MOV     #BIT13,RPM3(R4)  ;TO CLEAR THE 'SOT'
 2832  014212  004737  022014                JSR     PC,SAVRP        ;STORE THE RP11E REGISTERS
 2833  014216  013737  001234  001126        MOV     $RPDA,$BDDAT    ;STORE THE CONTENTS
 2834  014224  013737  001234  001160        MOV     $RPDA,$TMPD     ;MOVE CONTENTS TO A WORKING LOCATION
 2835  014232  042737  177417  001160        BIC     #^C360,$TMPD    ;MASK OUT ALL BUT THE 'SOT' BITS
 2836  014240  001401                        BEQ     1$
 2837  014242  104062                        ERROR   62              ;SOT DID NOT CLEAR WITH INDEX PULSE
 2838  014244  000004                1$:     SCOPE                   ;LOOP ?
 2839
 2840
 2841                                        ;;*****************************************************************
 2842                                        ;*TEST 50       SILO TEST, PART 1
 2843
 2844                                        ;*THIS TEST CHECKS THE SILO MEMORY IN MAINTENANCE
 2845                                        ;*MODE.  IF INREADY IS SET, DATA IS OUTPUT TO REGISTER
 2846                                        ;*'SILO' WHICH IS THE SILO MEMORY.  AFTER THE DATA FILTERS
 2847                                        ;*THRU THE MEMORY, OUTREADY GOES TRUE.  THE DATA IS READ
 2848                                        ;*BACK AND COMPARED.
 2849
 2850                                        ;;*****************************************************************
 2851  014246                        TST50:
 2852  014246  000240                        NOP
 2853  014250  012737  014270  001106        MOV     #SILOT,$LPADR   ;LOAD LOOP ON TEST ADDRESS
 2854  014256  012737  014270  001110        MOV     #SILOT,$LPERR   ;LOAD ERROR LOOP ADDRESS
```

```
 2855  014264  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2856  014270  012706  001100          SILOT:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
 2857  014274  012737  000001  001124          MOV     #1,$GDDAT       ;INITIALIZE FLOATING ONE PATTERN
 2858  014302  005005                          CLR     R5              ;PATTERN FLAG
 2859  014304  012737  014304  001110  1$:     MOV     #1$,$LPERR      ;RESTORE THE LOOP ON ERROR ADDRESS
 2860  014312  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
 2861  014316  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2862  014322  032737  004000  001236          BIT     #BIT11,$RPM1    ;IS INREADY SET?
 2863  014330  001002                          BNE     2$              ;BRANCH IF SET
 2864  014332  104064                          ERROR   64              ;SILO INREADY IS NOT SET AFTER CLEAR
 2865  014334  000454                          BR      9$              ;EXIT
 2866  014336  012737  014336  001110  2$:     MOV     #2$,$LPERR      ;CHANGE LOOP ON ERROR ADDRESS
 2867  014344  013764  001124  000026          MOV     $GDDAT,SILO(R4) ;LOAD DATA IN SILO
 2868  014352  012701  000060                  MOV     #60,R1          ;WAIT FOR OUT READY TO SET
 2869  014356  005301                  3$:     DEC     R1
 2870  014360  001376                          BNE     3$
 2871  014362  004737  022014          4$:     JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
 2872  014366  032737  010000  001236          BIT     #BIT12,$RPM1    ;IS OUTREADY SET?
 2873  014374  001002                          BNE     5$              ;BRANCH IF SET
 2874  014376  104065                          ERROR   65              ;SILO OUTREADY IS NOT SET
 2875  014400  000432                          BR      9$
 2876  014402  016437  000026  001126  5$:     MOV     SILO(R4),$BDDAT ;GET DATA BACK FROM SILO
 2877  014410  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS DATA CORRECT?
 2878  014416  001401                          BEQ     6$
 2879  014420  104066                          ERROR   66              ;DATA READ FROM SILO INCORRECT
 2880  014422  005705                  6$:     TST     R5              ;ARE WE FLOATING A ONE?
 2881  014424  001012                          BNE     8$              ;BRANCH IF ZERO
 2882  014426  006337  001124                  ASL     $GDDAT          ;SHIFT PATTERN
 2883  014432  103401                          BCS     7$              ;BRANCH IF PATTERN EXCEEDED
 2884  014434  000723                          BR      1$
 2885  014436  012705  000001          7$:     MOV     #1,R5           ;SET PATTERN FLAG
 2886  014442  012737  077777  001124          MOV     #77777,$GDDAT   ;FLOATING ZERO PATTERN
 2887  014450  000715                          BR      1$
 2888  014452  006237  001124          8$:     ASR     $GDDAT          ;SHIFT FLOATING ZERO
 2889  014456  052737  100000  001124          BIS     #BIT15,$GDDAT
 2890  014464  103707                          BCS     1$
 2891  014466  000004                  9$:     SCOPE                   ;LOOP ?
 2892
 2893
 2894                                          ;;*****************************************************************
 2895                                          ;*TEST 51        SILO TEST, PART 2
 2896
 2897                                          ;*ENSURE THAT THE SILO MEMORY CAN HOLD 64 DISCREET NUMBERS
 2898                                          ;*AT ONE TIME.  AFTER LOADING THE 64 NUMBERS SIGNAL INREADY
 2899                                          ;*SHOULD CLEAR INDICATING THE SILO IS FULL.  THE SILO IS THEN
 2900                                          ;*READ OUT EXPECTING SEQUENTIAL NUMBERS OF 1 THRU 100 OCTAL. AT
 2901                                          ;*THIS TIME OUT READY SHOULD CLEAR.
 2902
 2903                                          ;;*****************************************************************
 2904  014470                                  TST51:
 2905  014470  000240                                  NOP
 2906  014472  012737  014520  001106                  MOV     #SILOT1,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 2907  014500  012737  014520  001110                  MOV     #SILOT1,$LPERR  ;LOAD ERROR LOOP ADDRESS
 2908  014506  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
 2909  014512  012737  000012  001162                  MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
 2910  014520  012706  001100          SILOT1: MOV     #STACK,SP       ;SETUP THE STACK POINTER
```

```
2911  014524  012737  000001  001124          MOV     #1,$GDDAT       ;INITIALIZE TEST PATTERN
2912  014532  012737  014532  001110  1$:     MOV     #1$,$LPERR      ;RESTORE THE LOOP ON ERROR ADDRESS
2913  014540  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
2914  014544  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11E REGISTERS
2915  014550  032737  004000  001236          BIT     #BIT11,$RPM1    ;IS INREADY SET?
2916  014556  001001                          BNE     2$              ;BRANCH IF SET
2917  014560  104064                          ERROR   64              ;SILO INREADY SHOULD BE SET
2918  014562  012737  014562  001110  2$:     MOV     #2$,$LPERR      ;CHANGE THE LOOP ON ERROR ADDRESS
2919  014570  012706  001100                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
2920  014574  013764  001124  000026          MOV     $GDDAT,SILO(R4) ;LOAD PATTERN IN SILO
2921  014602  005237  001124                  INC     $GDDAT          ;UPDATE PATTERN
2922  014606  022737  000101  001124          CMP     #65.,$GDDAT     ;IS THE SILO FULL?
2923  014614  001362                          BNE     2$              ;BRANCH NOT FULL
2924  014616  016437  000016  001236          MOV     RPM1(R4),$RPM1  ;SAVE THE REGISTER
2925  014624  032737  004000  001236          BIT     #BIT11,$RPM1    ;DID INREADY CLEAR?
2926  014632  001401                          BEQ     3$              ;BRANCH IF YES
2927  014634  104067                          ERROR   67              ;SILO INREADY DID NOT CLEAR
2928  014636  012737  000001  001124  3$:     MOV     #1,$GDDAT       ;RESET PATTERN
2929  014644  016437  000016  001236  4$:     MOV     RPM1(R4),$RPM1  ;SAVE THE REGISTER
2930  014652  032737  010000  001236          BIT     #BIT12,$RPM1    ;IS OUTREADY SET?
2931  014660  001001                          BNE     5$              ;BRANCH IF SET
2932  014662  104065                          ERROR   65              ;SILO OUTREADY SHOULD BE SET
2933  014664  012737  015014  001164  5$:     MOV     #9$,$ESCAPE     ;SETUP THE ESCAPE ADDRESS
2934  014672  016437  000026  001126          MOV     SILO(R4),$BDDAT ;READ SILO MEMORY
2935  014700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS DATA CORRECT?
2936  014706  001405                          BEQ     6$              ;BRANCH IF EQUAL
2937  014710  104066                          ERROR   66              ;INCORRECT DATA RECEIVED FROM SILO
2938  014712  032777  000200  164220          BIT     #SW07,@SWR      ;IS SWITCH 7 SET ?
2939  014720  001022                          BNE     7$              ;BR IF IT IS - BYPASS REST OF TEST
2940  014722  012705  000400          6$:     MOV     #256.,R5        ;CONSTANT FOR STALL
2941  014726  005305                  10$:    DEC     R5              ;STALL AND WAIT FOR 'INPUT READY' TO
2942  014730  001376                          BNE     10$             ;SET (FOR THE FIRST TIME THROUGH THE LOOP)
2943  014732  016437  000016  001236          MOV     RPM1(R4),$RPM1  ;SAVE THE MAINTENANCE REGISTER
2944  014740  005237  001124                  INC     $GDDAT          ;UPDATE EXPECTED PATTERN
2945  014744  022737  000101  001124          CMP     #65.,$GDDAT     ;HAVE 64 WORDS BEEN READ
2946  014752  001413                          BEQ     8$              ;BRANCH IF NO
2947  014754  032737  004000  001236          BIT     #BIT11,$RPM1    ;IS INPUT READY SET ?
2948  014762  001001                          BNE     7$              ;BR IF IT IS
2949  014764  104110                          ERROR   110             ;SILO NOT FULL, INPUT READY SOULD BE SET
2950  014766  032737  010000  001236  7$:     BIT     #BIT12,$RPM1    ;SEE IF OUTPUT READY STILL SET
2951  014774  001333                          BNE     5$              ;BR IF OUTPUT READY SET
2952  014776  104111                          ERROR   111             ;WORDS STILL IN SILO, OUTPUT READY
2953                                                                  ;SHOULD BE SET
2954  015000  000731                          BR      5$              ;CONTINUE
2955  015002  032737  010000  001236  8$:     BIT     #BIT12,$RPM1    ;HAS SILO OUTREADY CLEARED ?
2956  015010  001401                          BEQ     9$              ;BR IF CLEAR
2957  015012  104070                          ERROR   70              ;SILO SHOULD BE EMPTY BUT
2958                                                                  ;OUTREADY IS STILL SET
2959  015014  005037  001164          9$:     CLR     $ESCAPE         ;CLEAR THE ESCAPE ADDRESS
2960  015020  000004                          SCOPE                   ;LOOP ?
2961
2962
2963                                  ;;**********************************************************************
2964                                  ;*TEST 52        TEST 'SEEK' COMMAND BUS SIGNALS
2965
2966                                  ;*TEST THE OPERATION OF A SEEK COMMAND IN MAINTENANCE MODE.
```

```
                                        ;*ISSUE A SEEK COMMAND AND CHECK THE SETTING OF SET CYLINDER
                                        ;*RESET HEAD, SET HEAD,SEEK START AND CAR BITS 0 THRU 7.
                                        ;*ALL THESE SIGNALS ARE FOUND ON BUS OUT CONTROL.

                                        ;;*****************************************************************
015022                                  TST52:
015022  000240                                  NOP
015024  012737  015044  001106                  MOV     #1$,$LPADR      ;LOAD LOOP ON TEST ADDRESS
015032  012737  015052  001110                  MOV     #SEEK,$LPERR    ;LOAD ERROR LOOP ADDRESS
015040  013704  001210                          MOV     RPADR,R4        ;RP11E BUS ADDRESS
015044  012706  001100          1$:     MOV     #STACK,SP       ;SETUP THE STACK POINTER
015050  005005                          CLR     R5              ;CLEAR PASS FLAG
015052  004737  021706          SEEK:   JSR     PC,CLRP         ;CLEAR THE CONTROLLER
015056  004737  022014                  JSR     PC,SAVRP        ;SAVE THE RP11 REGISTERS
015062  013737  001236  001126          MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
015070  042737  174000  001126          BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
015076  001403                          BEQ     2$              ;BRANCH IF RESULT IS ZERO
015100  005037  001124                  CLR     $GDDAT
015104  104071                          ERROR   71              ;SOME BUS OUT SIGNALS TO THE
                                                                ;DRIVE ARE SET AFTER RESET
015106  012764  000252  000012  2$:     MOV     #170.,RPCA(R4)  ;LOAD CYCL 170 INTO RPCA
015114  005705                          TST     R5
015116  001403                          BEQ     3$
015120  012764  000125  000012          MOV     #85.,RPCA(R4)   ;LOAD CYLINDER 85 INTO REGISTER
015126  112764  000017  000015  3$:     MOVB    #17,RPDA+1(R4)  ;LOAD TRACK ADDR
015134  005705                          TST     R5              ;IS THIS FIRST PASS
015136  001403                          BEQ     4$              ;BRANCH IF YES
015140  112737  000020  001235          MOVB    #20,$RPDA+1     ;SET HIGH ORDER BIT OF TRACK ADDR
015146  112764  000011  000004  4$:     MOVB    #11,RPCS(R4)    ;ISSUE SEEK COMMAND
015154  012737  060001  001206          MOV     #BIT14!BIT13!BIT00,TPL
015162  004037  021736                  JSR     R0,TIMEP        ;GENERATE 5 TIMING PULSES
015166  000005                          .WORD   5               ;CONSTANT FOR 5 TIMING PULSES
015170  004737  022014                  JSR     PC,SAVRP        ;GET THE REGISTERS
015174  013737  001236  001126          MOV     $RPM1,$BDDAT    ;GET CONTROL LINES FROM RPM1
015202  042737  174000  001126          BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
015210  012737  000525  001124          MOV     #525,$GDDAT     ;LOAD EXPECTED VALUE OF RPM1
015216  005705                          TST     R5              ;IS THIS FIRST PASS
015220  001403                          BEQ     5$              ;BRANCH IF YES
015222  012737  000652  001124          MOV     #652,$GDDAT
015230  023737  001124  001126  5$:     CMP     $GDDAT,$BDDAT   ;WERE CONTENTS OF RPM1 CORRECT?
015236  001401                          BEQ     6$              ;BRANCH IF YES
015240  104072                          ERROR   72              ;THE CONTROL SIGNAL SET CYLINDER
                                                                ;AND THE CONTENTS OF CAR SHOULD
                                                                ;BE ON THE BUS OUT LINES
015242  004037  021736          6$:     JSR     R0,TIMEP        ;GENERATE 4 CLOCK PULSES
015246  000004                          .WORD   4               ;CONSTANT FOR 4 CLOCK PULSES
015250  004737  022014                  JSR     PC,SAVRP        ;GET THE REGISTERS
015254  013737  001236  001126          MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
015262  042737  174000  001126          BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
015270  012737  002010  001124          MOV     #BIT10!BIT03,$GDDAT  ;LOAD EXPECTED VALUE OF RPM1
015276  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ARE CONTENTS OF RPM1 CORRECT?
015304  001401                          BEQ     7$              ;BRANCH IF YES
015306  104073                          ERROR   73              ;THE CONTROL SIGNAL RESET HEAD
                                                                ;SHOULD BE ON THE BUS OUT LINES
015310  004037  021736          7$:     JSR     R0,TIMEP        ;GENERATE 4 CLOCK PULSES
015314  000004                          .WORD   4               ;CONSTANT FOR 4 CLOCK PULSES
```

```
3023  015316  004737  022014                        JSR     PC,SAVRP        ;GET THE REGISTERS
3024  015322  013737  001236  001126                MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
3025  015330  042737  174000  001126                BIC     #174000,$BDDAT  ;CLEAR UNWANTED
3026  015336  012737  001360  001124                MOV     #1360,$GDDAT    ;LOAD EXPECTED VALUE OF RPM1
3027  015344  005705                                TST     R5              ;IS THIS THE FIRST PASS ?
3028  015346  001403                                BEQ     8$              ;BRANCH IF YES
3029  015350  012737  001010  001124                MOV     #1010,$GDDAT    ;SECOND PASS VALUE
3030  015356  023737  001124  001126  8$:           CMP     $GDDAT,$BDDAT   ;ARE CONTENTS OF RPM1 CORRECT?
3031  015364  001401                                BEQ     9$
3032  015366  104074                                ERROR   74              ;THE CONTROL SIGNAL SET HEAD AND
3033                                                                        ;THE HEAD ADDRESS SHOULD BE ON
3034                                                                        ;THE BUS OUT LINES
3035  015370  004037  021736          9$:           JSR     R0,TIMEP        ;GENERATE 4 CLOCK PULSES
3036  015374  000004                                .WORD   4               ;CONSTANT FOR 4 CLOCK PULSES
3037  015376  004737  022014                        JSR     PC,SAVRP        ;GET THE REGISTERS
3038  015402  013737  001236  001126                MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
3039  015410  042737  174000  001126                BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
3040  015416  012737  002004  001124                MOV     #BIT10!BIT02,$GDDAT  ;LOAD EXPECTED BITS
3041  015424  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;ARE CONTENTS OF RPM1 CORRECT?
3042  015432  001401                                BEQ     10$             ;BRANCH IF YES
3043  015434  104075                                ERROR   75              ;THE CONTROL SIGNAL SEEK START
3044                                                                        ;SHOULD BE ON THE BUS OUT LINES
3045  015436  005704                  10$:           TST     R4              ;IS THIS FIRST PASS?
3046  015440  001002                                BNE     11$             ;BRANCH IF NO
3047  015442  005205                                INC     R5              ;SET SECOND PASS INDICATOR
3048  015444  000602                                BR      SEEK            ;MAKE SECOND PASS
3049  015446  000004                  11$:           SCOPE                   ;LOOP ?
3050
3051
3052                                                ;;*****************************************************************
3053                                                ;*TEST 53       TEST 'HOME SEEK' COMMAND BUS SIGNALS
3054
3055                                                ;*ISSUE A RESTORE COMMAND AND CHECK THE GENERATION OF
3056                                                ;*THE SIGNAL 'RESTORE' ON THE BUS OUT CONTROL LOGIC.
3057
3058                                                ;;*****************************************************************
3059  015450                                        TST53:
3060  015450  000240                                        NOP
3061  015452  012737  015472  001106                        MOV     #RESTOR,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3062  015460  012737  015472  001110                        MOV     #RESTOR,$LPERR  ;LOAD ERROR LOOP ADDRESS
3063  015466  013704  001210                        MOV     RPADR,R4        ;RP11E BUS ADDRESS
3064  015472  012706  001100                RESTOR: MOV     #STACK,SP       ;SETUP THE STACK POINTER
3065  015476  004737  021706                        JSR     PC,CLRP         ;CLEAR THE CONTROLLER
3066  015502  004737  022014                        JSR     PC,SAVRP        ;SAVE THE REGISTERS
3067  015506  013737  001236  001126                MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
3068  015514  042737  174000  001126                BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
3069  015522  005737  001126                        TST     $BDDAT          ;ANY SET ?
3070  015526  001403                                BEQ     1$              ;BRANCH IF RESULT IS YES
3071  015530  005037  001124                        CLR     $GDDAT
3072  015534  104071                                ERROR   71              ;SOME BUS OUT SIGNALS TO THE
3073                                                                        ;DRIVE ARE SET AFTER RESET
3074  015536  012764  000015  000004  1$:           MOV     #15,RPCS(R4)    ;ISSUE HOME COMMAND
3075  015544  012737  060001  001206                MOV     #BIT14!BIT13!BIT00,TPL
3076  015552  004037  021736                        JSR     R0,TIMEP        ;GENERATE 4 CLOCK PULSES
3077  015556  000004                                .WORD   4               ;CONSTANT FOR 4 CLOCK PULSES
3078  015560  004737  022014                        JSR     PC,SAVRP        ;GET THE REGISTERS
```

```
3079  015564  013737  001236  001126          MOV     $RPM1,$BDDAT    ;GET CONTROL LINES FROM RPM1
3080  015572  042737  174000  001126          BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
3081  015600  012737  002100  001124          MOV     #BIT10!BIT06,$GDDAT  ;LOAD EXPECTED VALUE OF RPM1
3082  015606  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;WERE CONTENTS OF RPM1 CORRECT?
3083  015614  001401                          BEQ     2$              ;BRANCH IF YES
3084  015616  104076                          ERROR   76              ;ISSUED HOME COMMAND-EXPECTED
3085                                                                  ;RESTORE AND CONTROL TO BE SET
3086                                                                  ;ON BUS OUT LINES
3087  015620  000004                  2$:     SCOPE                   ;LOOP ?
3088
3089
3090                                  ;;*****************************************************************
3091                                  ;*TEST 54        TEST 'READ' BUS SIGNALS
3092
3093                                  ;*ISSUE A READ COMMAND WITH NO SEEK IMPLIED AND CHECK THE GENERATION
3094                                  ;*OF THE SIGNAL READ IN THE BUS OUT LOGIC.
3095
3096                                  ;;*****************************************************************
3097  015622                          TST54:
3098  015622  000240                          NOP
3099  015624  012737  015644  001106          MOV     #READT,$LPADR   ;LOAD LOOP ON TEST ADDRESS
3100  015632  012737  015644  001110          MOV     #READT,$LPERR   ;LOAD ERROR LOOP ADDRESS
3101  015640  013704  001210                  MOV     RPADR,R4        ;RP11E BUS ADDRESS
3102  015644  012706  001100          READT:  MOV     #STACK,SP       ;SETUP THE STACK POINTER
3103  015650  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
3104  015654  012764  177777  000006          MOV     #-1,RPWC(R4)    ;LOAD WORD COUNT
3105  015662  012764  000017  000004          MOV     #17,RPCS(R4)    ;ISSUE READ COMMAND
3106  015670  012737  060001  001206          MOV     #BIT14!BIT13!BIT00,TPL
3107  015676  004037  021736                  JSR     R0,TIMEP        ;GENERATE 4 CLOCK PULSES
3108  015702  000004                          .WORD   4               ;CONSTANT FOR 4 CLOCK PULSES
3109  015704  012737  060401  001206          MOV     #BIT14!BIT13!BIT08!BIT00,TPL
3110  015712  004037  021736                  JSR     R0,TIMEP        ;GENERATE 3 CLOCK PULSES
3111  015716  000003                          .WORD   3               ;CONSTANT FOR 3 CLOCK PULSES
3112  015720  004737  022014                  JSR     PC,SAVRP        ;GET THE REGISTERS
3113  015724  013737  001236  001126          MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
3114  015732  042737  174000  001126          BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
3115  015740  012737  002042  001124          MOV     #BIT10!BIT05!BIT01,$GDDAT  ;LOAD EXPECTED VALUE OF RPM1
3116  015746  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ARE CONTENTS OF RPM1 CORRECT
3117  015754  001401                          BEQ     1$              ;BRANCH IF YES
3118  015756  104077                          ERROR   77              ;ISSUED READ WITH NO SEEK COMMAND
3119                                                                  ;EXPECTED CONTROL, SELECT HEAD, AND READ
3120                                                                  ;SIGNALS ON THE BUS OUT LINES
3121  015760  000004                  1$:     SCOPE                   ;LOOP ?
3122
3123
3124                                  ;;*****************************************************************
3125                                  ;*TEST 55        TEST 'WRITE' BUS SIGNALS
3126
3127                                  ;*ISSUE A WRITE FORMAT COMMAND AND CHECK THE GENERATION OF THE
3128                                  ;*WRITE SIGNAL IN THE BUS OUT LOGIC.
3129
3130                                  ;;*****************************************************************
3131  015762                          TST55:
3132  015762  000240                          NOP
3133  015764  012737  016004  001106          MOV     #WRITE,$LPADR   ;LOAD LOOP ON TEST ADDRESS
3134  015772  012737  016004  001110          MOV     #WRITE,$LPERR   ;LOAD ERROR LOOP ADDRESS
```

```
3135  016000  013704  001210          WRITE:  MOV     RPADR,R4        ;RP11E BUS ADDRESS
3136  016004  012706  001100                  MOV     #STACK,SP       ;SETUP THE STACK POINTER
3137  016010  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
3138  016014  012764  177777  000006          MOV     #-1,RPWC(R4)    ;SET UP WORD COUNT
3139  016022  012764  000001  000014          MOV     #1,RPDA(R4)     ;SELECT SECTOR 1, TRACK 0
3140  016030  012764  014003  000004          MOV     #14003,RPCS(R4) ;ISSUE WRITE FORMAT COMMAND
3141  016036  012737  060001  001206          MOV     #BIT14!BIT13!BIT00,TF
3142  016044  004037  021736                  JSR     R0,TIMEP        ;GENERATE 19 CLOCK PULSES
3143  016050  000023                          .WORD   19.             ;CONSTANT FOR 19 CLOCK PULSES
3144  016052  004037  021766                  JSR     R0,INDEXP       ;GENERATE 1 INDEX PULSE
3145  016056  000001                          .WORD   1               ;CONSTANT FOR 1 INDEX PULSE
3146  016060  012764  020400  000022          MOV     #BIT13!BIT08,RPM3(R4)   ;GENERATE SECTOR PULSE
3147  016066  012764  020000  000022          MOV     #BIT13,RPM3(R4)
3148  016074  012764  020400  000022          MOV     #BIT13!BIT08,RPM3(R4)   ;GENERATE SECTOR PULSE
3149  016102  004037  021736                  JSR     R0,TIMEP        ;GENERATE 3 CLOCK PULSES
3150  016106  000003                          .WORD   3               ;CONSTANT FOR 3 CLOCK PULSES
3151  016110  004737  022014                  JSR     PC,SAVRP        ;GET THE REGISTERS
3152  016114  013737  001236  001126          MOV     $RPM1,$BDDAT    ;GET CONTENTS OF RPM1
3153  016122  042737  174000  001126          BIC     #174000,$BDDAT  ;CLEAR UNWANTED BITS
3154  016130  012737  002061  001124          MOV     #BIT10!BIT05!BIT04!BIT00,$GDDAT  ;LOAD EXPECTED VALUE OF RPM1
3155  016136  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;ARE CONTENTS OF RPM1 CORRECT
3156  016144  001401                          BEQ     1$              ;BRANCH IF YES
3157  016146  104100                          ERROR   100             ;ISSUED A WRITE FORMAT COMMAND
3158                                                                  ;AND EXPECTED CONTROL, SELECT HEADS, ERASE,
3159                                                                  ;AND WRITE SIGNALS ON THE BUS OUT LOGIC
3160  016150  000004                  1$:     SCOPE                   ;LOOP ?
3161  016152  004737  021706                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
3162  016156  000137  016162                  JMP     $EOP            ;GO TO THE END OF PASS ROUTINE
3163
3164
3165                                          .SBTTL  END OF PASS ROUTINE
3166
3167                                          ;;*****************************************************************
3168                                          ;*INCREMENT THE PASS NUMBER ($PASS)
3169                                          ;*INDICATE END-OF-PROGRAM AFTER 8. PASSES THRU THE PROGRAM
3170                                          ;*IF THERES A MONITOR GO TO IT
3171                                          ;*IF THERE ISN'T JUMP TO RSTART
3172
3173  016162                          $EOP:
3174  016162  104401  016170                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3175  016166  000410                          BR      64$             ;;GET OVER THE ASCIZ
3176                                  ;;65$:  .ASCIZ  <15><12><12>/END OF PASS/
3177  016210                          64$:
3178  016210  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
3179  016214  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3180  016216     005                          .BYTE   5               ;;TYPE 5 DIGIT(S)
3181  016217     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
3182  016220  104401  016226                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3183  016224  000413                          BR      66$             ;;GET OVER THE ASCIZ
3184                                  ;;67$:  .ASCIZ  /   ERRORS DETECTED = /
3185  016254                          66$:
3186  016254  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
3187  016260  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3188  016262  005037  001112                  CLR     $ERTTL          ;CLEAR THE ERROR TOTAL
3189  016266  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
3190  016272  005037  001162                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
```

```
                                                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
                                                    BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
                                                    DEC     (PC)+           ;;LOOP?
                                    SEOPCT:         .WORD   0.
                                                    BGT     $DOAGN          ;;YES
                                                    MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
                                    SENDCT:         .WORD   0.
                                                    $EOPCT
                                                    TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                    BR      64$             ;;GET OVER THE ASCIZ
                                    ;;65$:          .ASCIZ  <15><12>/END OF TEST/
                                    64$:
                                                    TYPE    ,$ENULL         ;TYPE NULL CHARACTER
                                    $GET42:         MOV     @#42,R0         ;;GET MONITOR ADDRESS
                                                    BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
                                                    RESET                   ;;CLEAR THE WORLD
                                    $ENDAD:         JSR     PC,(R0)         ;;GO TO MONITOR
                                                    NOP                     ;;SAVE ROOM
                                                    NOP                     ;;FOR
                                                    NOP                     ;;ACT11
                                    $DOAGN:
                                                    JMP     @(PC)+          ;;RETURN
                                    $RTNAD:         .WORD   RSTART
                                    $ENULL:         .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                                    .EVEN

                                                    .SBTTL  **** SUBROUTINES ****

                                                    .SBTTL  ERROR HANDLER ROUTINE

                                                    ;;**************************************************************
                                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                                    ;*AND GO TO TYPERR ON ERROR
                                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                                    ;*SW15=1        HALT ON ERROR
                                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                                    ;*SW10=1        BELL ON ERROR
                                                    ;*SW09=1        LOOP ON ERROR
                                                    ;*CALL
                                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

                                    $ERROR:
                                                    CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
                                    7$:             INCB    $ERFLG          ;;SET THE ERROR FLAG
                                                    BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
                                                    MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
                                                    BIT     #BIT10,@SWR     ;;BELL ON ERROR?
                                                    BEQ     1$              ;;NO - SKIP
                                                    TYPE    ,$BELL          ;;RING BELL
                                    1$:             INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
                                                    MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
                                                    SUB     #2,$ERRPC
                                                    MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
                                                    BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
                                                    BNE     20$             ;;SKIP TYPEOUTS
```

```
3247  016472  004737  016556              JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
3248  016476  104401  001173              TYPE    ,$CRLF
3249  016502                        20$:
3250  016502  005777  162432        2$:   TST     @SWR            ;;HALT ON ERROR
3251  016506  100002                      BPL     3$              ;;SKIP IF CONTINUE
3252  016510  000000                      HALT                    ;;HALT ON ERROR!
3253  016512  104407                      CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
3254  016514  032777  001000  162416  3$:   BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
3255  016522  001402                      BEQ     4$              ;;BR IF NO
3256  016524  013716  001110              MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
3257  016530  005737  001164        4$:   TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
3258  016534  001402                      BEQ     5$              ;;BR IF NONE
3259  016536  013716  001164              MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
3260  016542                        5$:
3261  016542  022737  016364  000042      CMP     #SENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
3262  016550  001001                      BNE     6$              ;;BRANCH IF NO
3263  016552  000000                      HALT                    ;;YES
3264  016554                        6$:
3265  016554  000002                      RTI                     ;;RETURN
3266
3267  016556  104413              TYPERR: SAVREG                  ;SAVE R0-R5
3268  016560  005037  001160              CLR     $TMP0           ;CLEAR LOCATION FOR TEST NUMBER
3269  016564  113737  001102  001160      MOVB    $TSTNM,$TMP0    ;LOAD TEST NUMBER FOR TYPEOUT
3270  016572  005000                      CLR     R0              ;CLEAR R0 FOR ERROR NUMBER
3271  016574  113700  001114              MOVB    $ITEMB,R0       ;ERROR NUMBER
3272  016600  005300                      DEC     R0              ;FORM INDEX FOR ERROR TABLE
3273  016602  006300                      ASL     R0
3274  016604  006300                      ASL     R0
3275  016606  006300                      ASL     R0
3276  016610  062700  001244        1$:   ADD     #SERRTB,R0      ;FORM ADDRESS
3277  016614  012037  016630              MOV     (R0)+,2$        ;GET ERROR MESSAGE (EM) POINTER
3278  016620  001404                      BEQ     3$              ;BRANCH IF THERE ISN'T ONE
3279  016622  104401  001173              TYPE    ,$CRLF          ;"CARRIAGE RETURN - LINE FEED
3280  016626  104401                      TYPE
3281  016630  000000                2$:   .WORD   0               ;"EM" POINTER GOES HERE
3282  016632  012037  016646        3$:   MOV     (R0)+,4$        ;PICK UP DATA HEADER (DH) POINTER
3283  016636  001404                      BEQ     5$              ;BRANCH IF NONE
3284  016640  104401  001173              TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
3285  016644  104401                      TYPE
3286  016646  000000                4$:   .WORD   0               ;"DH" POI TER GOES HERE
3287  016650  012001                5$:   MOV     (R0)+,R1        ;PICKUP DATA TABLE (DT) POINTER
3288  016652  001430                      BEQ     20$             ;BRANCH IF NONE
3289  016654  005005                      CLR     R5              ;SET INDENT SWITCH
3290  016656  012000                      MOV     (R0)+,R0        ;DATA FORMAT (DF) POINTER
3291  016660  012002                      MOV     (R0)+,R2        ;NUMBER OF DH'S TO TYPE
3292  016662  001441                      BEQ     17$             ;BRANCH IF DH NUMBER IS 0
3293  016664  005105                      COM     R5              ;NO INDENT
3294  016666  104401  001173              TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
3295  016672  112003                10$:  MOVB    (R0)+,R3        ;NUMBER OF DATA WORDS TO TYPE
3296  016674  112004                      MOVB    (R0)+,R4        ;AND HOW TO TYPE THEM
3297  016676  006004                11$:  ROR     R4              ;OCTAL OR DECIMAL?
3298  016700  103403                      BCS     12$             ;DECIMAL--BRANCH
3299  016702  013146                      MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
3300  016704  104402                      TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
3301  016706  000402                      BR      13$
3302  016710                        12$:
```

```
3303  016710  013146                          MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
3304  016712  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3305  016714  005303                  13$:    DEC     R3              ;MORE NUMBERS TO TYPE?
3306  016716  001403                          BEQ     14$             ;NO--BRANCH
3307  016720  104401  022112                  TYPE    ,BLNKS2         ;YES--TYPE SEPARATORS
3308  016724  000764                          BR      11$             ;LOOP
3309  016726  005302                  14$:    DEC     R2              ;MORE DH'S?
3310  016730  003421                          BLE     20$             ;NO--BRANCH
3311  016732  104401  001173                  TYPE    ,$CRLF          ;YES--START A NEW LINE
3312  016736  005105                          COM     R5              ;INDENT?
3313  016740  001002                          BNE     15$             ;NO--BRANCH
3314  016742  104401  022112                  TYPE    ,BLNKS2         ;YES--TYPE SPACES
3315  016746  012037  016754          15$:    MOV     (R0)+,16$       ;GET NEXT DH
3316  016752  104401                          TYPE                    ;AND TYPE IT
3317  016754  000000                  16$:    .WORD   0               ;DH POINTER GOES HERE
3318  016756  104401  001173                  TYPE    ,$CRLF          ;CARRIAGE RETURN-LINE FEED
3319  016762  005705                          TST     R5              ;INDENT?
3320  016764  001342                          BNE     10$             ;NO--BRANCH
3321  016766  104401  022112          17$:    TYPE    ,BLNKS2         ;YES--TYPE SPACES
3322  016772  000737                          BR      10$             ;LOOP
3323  016774  104414                  20$:    RESREG                  ;RESTORE R0-R5
3324  016776  000207                          RTS     PC              ;RETURN
3325
3326
3327                                  .SBTTL  TYPE ROUTINE
3328
3329                                  ;;**************************************************************
3330                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
3331                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3332                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3333                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3334                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3335                                  ;*
3336                                  ;*CALL:
3337                                  ;*1) USING A TRAP INSTRUCTION
3338                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3339                                  ;*OR
3340                                  ;*      TYPE
3341                                  ;*      MESADR
3342                                  ;*
3343
3344  017000  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
3345  017004  100002                  BPL     1$              ;;BR IF YES
3346  017006  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
3347  017010  000407                  BR      3$              ;;LEAVE
3348  017012  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
3349  017014  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
3350  017020  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
3351  017022  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
3352  017024  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
3353  017026  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
3354  017030  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
3355  017034  000002                  RTI                     ;;RETURN
3356  017036  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
3357  017042  001430                  BEQ     8$
3358  017044  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
```

```
3359  017050  001006                        BNE     5$
3360  017052  005726                        TST     (SP)+           ;;POP  (CR)(LF) EQUIV
3361  017054  104401                        TYPE                    ;;TYPE A CR AND LF
3362  017056  001173                        $CRLF
3363  017060  105037  017214                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
3364  017064  000755                        BR      2$              ;;GET NEXT CHARACTER
3365  017066  004737  017150        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
3366  017072  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
3367  017076  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
3368  017100  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
3369                                                                ;;AND THE NULL CHAR.
3370  017104  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
3371  017110  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
3372  017112  004737  017150                JSR     PC,$TYPEC       ;;GO TYPE A NULL
3373  017116  105337  017214                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
3374  017122  000770                        BR      7$              ;;LOOP
3375
3376                                ;HORIZONTAL TAB PROCESSOR
3377
3378  017124  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3379  017130  004737  017150        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
3380  017134  132737  000007  017214        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3381  017142  001372                        BNE     9$              ;;TAB STOP
3382  017144  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
3383  017146  000724                        BR      2$              ;;GET NEXT CHARACTER
3384  017150  105777  161774        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
3385  017154  100375                        BPL     $TYPEC
3386  017156  116677  000002  161766        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3387  017164  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3388  017172  001003                        BNE     1$              ;;BRANCH IF NO
3389  017174  105037  017214                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3390  017200  000406                        BR      $TYPEX          ;;EXIT
3391  017202  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3392  017210  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
3393  017212  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
3394  017214  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
3395  017216  000207        $TYPEX: RTS     PC
3396
3397
3398                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3399
3400                                ;;**************************************************************
3401                                ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3402                                ;;*OCTAL (ASCII) NUMBER AND TYPE IT.
3403                                ;;*$TYPOS--ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3404                                ;;*CALL:
3405                                ;;*     MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3406                                ;;*     TYPOS                   ;;CALL FOR TYPEOUT
3407                                ;;*     .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3408                                ;;*     .BYTE   M               ;;M=1 OR 0
3409                                ;;*                                     ;;1=TYPE LEADING ZEROS
3410                                ;;*                                     ;;0=SUPPRESS LEADING ZEROS
3411                                ;;*
3412                                ;;*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3413                                ;;*$TYPOS OR $TYPOC
3414                                ;;*CALL:
```

```
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
017220  017646  000000          STYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
017224  116637  000001  017443          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
017232  112637  017445                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
017236  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
017242  000406                          BR      $TYPON
017244  112737  000001  017443  STYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
017252  112737  000006  017445          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
017260  112737  000005  017442  STYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
017266  010346                          MOV     R3,-(SP)        ;;SAVE R3
017270  010446                          MOV     R4,-(SP)        ;;SAVE R4
017272  010546                          MOV     R5,-(SP)        ;;SAVE R5
017274  113704  017445                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
017300  005404                          NEG     R4
017302  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
017306  110437  017444                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
017312  113704  017443                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
017316  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
017322  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
017324  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
017326  000404                          BR      3$              ;;GO DO MSB
017330  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
017332  006105                          ROL     R5
017334  006105                          ROL     R5
017336  010503                          MOV     R5,R3
017340  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
017342  105337  017444                  DECB    $OMODE          ;;TYPE THIS DIGIT?
017346  100016                          BPL     7$              ;;BR IF NO
017350  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
017354  001002                          BNE     4$              ;;TEST FOR 0
017356  005704                          TST     R4              ;;SUPPRESS THIS 0?
017360  001403                          BEQ     5$              ;;BR IF YES
017362  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
017364  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
017370  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
017374  110337  017440                  MOVB    R3,8$           ;;SAVE FOR TYPING
017400  104401  017440                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
017404  105337  017442          7$:     DECB    $OCNT           ;;COUNT BY 1
017410  003347                          BGT     2$              ;;BR IF MORE TO DO
017412  002402                          BLT     6$              ;;BR IF DONE
017414  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
017416  000744                          BR      2$              ;;GO DO THE LAST DIGIT
017420  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
017422  012604                          MOV     (SP)+,R4        ;;RESTORE R4
017424  012603                          MOV     (SP)+,R3        ;;RESTORE R3
017426  016665  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
017434  012616                          MOV     (SP)+,(SP)
017436  000002                          RTI                     ;;RETURN
017440  000000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
```

```
3471  017441  000                     .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3472  017442  000             $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3473  017443  000             $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3474  017444  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3475
3476                          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3477
3478                          ;;**************************************************** ****
3479                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO  5-DIGIT
3480                          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
3481                          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
3482                          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
3483                          ;*REPLACED WITH SPACES.
3484                          ;*CALL:
3485                          ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
3486                          ;*      TYPOS                   ;;GO TO THE ROUTINE
3487
3488  017446                  $TYPOS:
3489  017446  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3490  017450  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3491  017452  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3492  017454  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3493  017456  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
3494  017460  012746  020200          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
3495  017464  016605  000020          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
3496  017470  100004                  BPL     1$              ;;BR IF INPUT IS POS.
3497  017472  005405                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
3498  017474  112766  000055  000001  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
3499  017502  005000          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
3500  017504  012703  017662          MOV     #$OBLK,R3       ;;SETUP THE OUTPUT POINTER
3501  017510  112723  000040          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
3502  017514  005002          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
3503  017516  016001  017652          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
3504  017522  160105          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
3505  017524  002402                  BLT     4$              ;;BR IF DONE
3506  017526  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
3507  017530  000774                  BR      3$
3508  017532  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
3509  017534  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
3510  017536  001002                  BNE     5$              ;;FALL THROUGH IF 0
3511  017540  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
3512  017542  100407                  BMI     7$              ;;BR IF YES
3513  017544  106316          5$:     ASLB    (SP)            ;;MSD?
3514  017546  103003                  BCC     6$              ;;BR IF NO
3515  017550  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
3516  017556  052702  000060  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
3517  017562  052702  000040  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
3518  017566  110223                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
3519  017570  005720                  TST     (R0)+           ;;JUST INCREMENTING
3520  017572  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
3521  017576  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
3522  017600  003002                  BGT     8$              ;;GO TO EXIT
3523  017602  010502                  MOV     R5,R2           ;;GET THE LSD
3524  017604  000764                  BR      6$              ;;GO CHANGE TO ASCII
3525  017606  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
3526  017610  100003                  BPL     9$              ;;BR IF NO
```

```
3527  017612  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
3528  017620  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
3529  017622  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
3530  017624  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
3531  017626  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3532  017630  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
3533  017632  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3534  017634  104401  017662                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
3535  017640  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
3536  017646  012616                          MOV     (SP)+,(SP)
3537  017650  000002                          RTI                     ;;RETURN TO USER
3538  017652  023420                  $DTBL:  10000.
3539  017654  001750                          1000.
3540  017656  000144                          100.
3541  017660  000012                          10.
3542  017662  000004                  $DBLK:  .BLKW   4
3543
3544                                  .SBTTL  TTY INPUT ROUTINE
3545
3546                                  ;;**********************************************************************
3547                                  .ENABL  LSB
3548
3549                                  ;;**********************************************************************
3550                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
3551                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
3552                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
3553                                  ;*WHEN OPERATING IN TTY FLAG MODE.
3554  017672  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
3555  017700  001074                          BNE     15$             ;;BRANCH IF NO
3556  017702  105777  161236                  TSTB    @$TKS           ;;CHAR THERE?
3557  017706  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
3558  017710  117746  161232                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
3559  017714  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
3560  017720  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
3561  017724  001062                          BNE     15$             ;;NO, RETURN TO USER
3562  017726  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
3563  017734  001456                          BEQ     15$             ;;BRANCH IF YES
3564
3565  017736  104401  020544                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
3566  017742  104401  020551          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
3567  017746  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
3568  017752  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3569  017754  104401  020562                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
3570  017760  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
3571  017762  005046                          CLR     -(SP)           ;;THE NEW SWR
3572  017764  105777  161154          7$:     TSTB    @$TKS           ;;CHAR THERE?
3573  017770  100375                          BPL     7$              ;;IF NOT TRY AGAIN
3574
3575  017772  117746  161150                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
3576  017776  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
3577
3578
3579
3580  020002  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
3581  020006  001005                          BNE     10$             ;;BRANCH IF NOT
3582  020010  104401  020537                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
```

```
3583  020014  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
3584  020020  000757                  BR      19$                     ;;LET'S TRY IT AGAIN
3585
3586
3587  020022  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
3588  020026  001022                  BNE     16$                     ;;BRANCH IF NO
3589  020030  005766  000004          TST     4(SP)                   ;;YES, IS IT THE FIRST CHAR?
3590  020034  001403                  BEQ     11$                     ;;BRANCH IF YES
3591  020036  016677  000002  161074  MOV     2(SP),@SWR              ;;SAVE NEW SWR
3592  020044  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
3593  020050  104401  001173          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
3594  020054  123727  001135  000001  CMPB    $INTAG,#1               ;;RE-ENABLE TTY KBD INTERRUPTS?
3595  020062  001003                  BNE     15$                     ;;BRANCH IF NOT
3596  020064  012777  000100  161052  MOV     #100,@STKS              ;;RE-ENABLE TTY KBD INTERRUPTS
3597  020072  000002                  15$:    RTI                     ;;RETURN
3598  020074  004737  017150          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
3599  020100  021627  000060          CMP     (SP),#60                ;;CHAR < 0?
3600  020104  002420                  BLT     18$                     ;;BRANCH IF YES
3601  020106  021627  000067          CMP     (SP),#67                ;;CHAR > 7?
3602  020112  003015                  BGT     18$                     ;;BRANCH IF YES
3603  020114  042726  000060          BIC     #60,(SP)+               ;;STRIP-OFF ASCII
3604  020120  005766  000002          TST     2(SP)                   ;;IS THIS THE FIRST CHAR
3605  020124  001403                  BEQ     17$                     ;;BRANCH IF YES
3606  020126  006316                  ASL     (SP)                    ;;NO, SHIFT PRESENT
3607  020130  006316                  ASL     (SP)                    ;;   CHAR OVER TO MAKE
3608  020132  006316                  ASL     (SP)                    ;;   ROOM FOR NEW ONE.
3609  020134  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
3610  020140  056616  177776          BIS     -2(SP),(SP)             ;;SET IN NEW CHAR
3611  020144  000707                  BR      7$                      ;;GET THE NEXT ONE
3612  020146  104401  001172          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
3613  020152  000720                  BR      20$                     ;;SIMULATE CONTROL-U
3614                                  .DSABL  LSB
3615
3616
3617                                  ;;*****************************************************************
3618                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3619                                  ;*CALL:
3620                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
3621                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
3622                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
3623                                  ;
3624
3625  020154  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
3626  020156  016666  000004  000002  MOV     4(SP),2(SP)             ;;SAVE THE PS
3627  020164  105777  160754          1$:     TSTB    @STKS           ;;WAIT FOR
3628  020170  100375                  BPL     1$                      ;;A CHARACTER
3629  020172  117766  160750  000004  MOVB    @STKB,4(SP)             ;;READ THE TTY
3630  020200  042766  177600  000004  BIC     #^C<177>,4(SP)          ;;GET RID OF JUNK IF ANY
3631  020206  026627  000004  000023  CMP     4(SP),#23               ;;IS IT A CONTROL-S?
3632  020214  001013                  BNE     3$                      ;;BRANCH IF NO
3633  020216  105777  160722          2$:     TSTB    @STKS           ;;WAIT FOR A CHARACTER
3634  020222  100375                  BPL     2$                      ;;LOOP UNTIL ITS THERE
3635  020224  117746  160716          MOVB    @STKB,-(SP)             ;;GET CHARACTER
3636  020230  042716  177600          BIC     #^C177,(SP)             ;;MAKE IT 7-BIT ASCII
3637  020234  022627  000021          CMP     (SP)+,#21               ;;IS IT A CONTROL-Q?
3638  020240  001366                  BNE     2$                      ;;IF NOT DISCARD IT
```

```
3639  020242  000750                        BR      1$              ;;YES, RESUME
3640  020244  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
3641  020252  002407                        BLT     4$              ;;BRANCH IF YES
3642  020254  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
3643  020262  003003                        BGT     4$              ;;BRANCH IF YES
3644  020264  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
3645  020272  000002              4$:       RTI                     ;;GO BACK TO USER
3646                                ;;**************************************************************
3647                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
3648                                ;*CALL:
3649                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
3650                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3651                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
3652
3653  020274  010346              $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
3654  020276  005046                      CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
3655  020300  012703  020530        1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
3656  020304  022703  020537        2$:   CMP     #$TTYIN+7,R3    ;;BUFFER FULL?
3657  020310  101456                      BLOS    4$              ;;BR IF YES
3658  020312  104410                      RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
3659  020314  112613                      MOVB    (SP)+,(R3)      ;;GET CHARACTER
3660  020316  122713  000177        10$:  CMPB    #177,(R3)       ;;IS IT A RUBOUT
3661  020322  001022                      BNE     5$              ;;BR IF NO
3662  020324  005716                      TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
3663  020326  001007                      BNE     6$              ;;BR IF NO
3664  020330  112737  000134  020526      MOVB    #'\,9$          ;;TYPE A BACK SLASH
3665  020336  104401  020526              TYPE    ,9$
3666  020342  012716  177777              MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
3667  020346  005303              6$:       DEC     R3              ;;BACKUP BY ONE
3668  020350  020327  020530              CMP     R3,#$TTYIN      ;;STACK EMPTY?
3669  020354  103434                      BLO     4$              ;;BR IF YES
3670  020356  111337  020526              MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
3671  020362  104401  020526              TYPE    ,9$             ;;GO TYPE
3672  020366  000746                      BR      2$              ;;GO READ ANOTHER CHAR.
3673  020370  005716              5$:       TST     (SP)            ;;RUBOUT KEY SET?
3674  020372  001406                      BEQ     7$              ;;BR IF NO
3675  020374  112737  000134  020526      MOVB    #'\,9$          ;;TYPE A BACK SLASH
3676  020402  104401  020526              TYPE    ,9$
3677  020406  005016                      CLR     (SP)            ;;CLEAR THE RUBOUT KEY
3678  020410  122713  000025        7$:   CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
3679  020414  001003                      BNE     8$              ;;BR IF NO
3680  020416  104401  020537              TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
3681  020422  000726                      BR      1$              ;;GO START OVER
3682  020424  122713  000022        8$:   CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
3683  020430  001011                      BNE     3$              ;;BRANCH IF NO
3684  020432  105013                      CLRB    (R3)            ;;CLEAR THE CHARACTER
3685  020434  104401  001173              TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
3686  020440  104401  020530              TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
3687  020444  000717                      BR      2$              ;;GO PICKUP ANOTHER CHACTER
3688  020446  104401  001172        4$:   TYPE    ,$QUES          ;;TYPE A '?'
3689  020452  000712                      BR      1$              ;;CLEAR THE BUFFER AND LOOP
3690  020454  111337  020526        3$:   MOVB    (R3),9$         ;;ECHO THE CHARACTER
3691  020460  104401  020526              TYPE    ,9$
3692  020464  122723  000015              CMPB    #15,(R3)+       ;;CHECK FOR RETURN
3693  020470  001305                      BNE     2$              ;;LOOP IF NOT RETURN
3694  020472  105063  177777              CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
```

```
3695  020476  104401  001174                  TYPE    $LF             ;;TYPE A LINE FEED
3696  020502  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
3697  020504  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3698  020506  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
3699  020510  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
3700  020516  012766  020530  000004          MOV     #$TTYIN,4(SP)
3701  020524  000002                          RTI                     ;;RETURN
3702  020526     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
3703  020527     000                          .BYTE   0               ;;TERMINATOR
3704  020530  000007                  $TTYIN: .BLKB   7               ;;RESERVE 7 BYTES FOR TTY INPUT
3705  020537     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
3706  020544  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
3707  020551     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
3708  020556  036440  000040
3709  020562  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
3710  020570  020075     000
3711          020574                  .EVEN
3712
3713                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
3714
3715                                  ;;**************************************************************
3716                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
3717                                  ;*CHANGE IT TO BINARY.
3718                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
3719                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
3720                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
3721                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
3722                                  ;*CALL:
3723                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
3724                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
3725                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
3726
3727  020574  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
3728  020576  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
3729  020604  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3730  020606  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3731  020610  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3732  020612  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
3733  020614  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
3734  020616  010037  020722                  MOV     R0,5$           ;;AND SAVE IT
3735  020622  005001                          CLR     R1              ;;CLEAR DATA WORD
3736  020624  005002                          CLR     R2
3737  020626  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
3738  020630  001420                          BEQ     3$              ;;IF ZERO GET OUT
3739  020632  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
3740  020636  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
3741  020640  122716  000067                  CMPB    #'7,(SP)
3742  020644  002423                          BLT     4$
3743  020646  006301                          ASL     R1              ;;*2
3744  020650  006102                          ROL     R2
3745  020652  006301                          ASL     R1              ;;*4
3746  020654  006102                          ROL     R2
3747  020656  006301                          ASL     R1              ;;*8
3748  020660  006102                          ROL     R2
3749  020662  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
3750  020666  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
```

```
3751  020670  000756                          BR      2$              ;;LOOP
3752  020672  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
3753  020674  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
3754  020700  010237  020732                  MOV     R2,$HIOCT
3755  020704  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3756  020706  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
3757  020710  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3758  020712  000002                          RTI                     ;;RETURN
3759  020714  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
3760  020716  105010                          CLRB    (R0)            ;;SET A TERMINATOR
3761  020720  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
3762  020722  000000                  5$:     .WORD   0
3763  020724  104401  001172                  TYPE    $QUES           ;;"?" "CR" & "LF"
3764  020730  000730                          BR      1$              ;;TRY AGAIN
3765  020732  000000                  $HIOCT: .WORD,  0               ;;HIGH ORDER BITS GO HERE
3766
3767                                  .SBTTL  SCOPE HANDLER ROUTINE
3768
3769                                  ;;**************************************************************
3770                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3771                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3772                                  ;*AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
3773                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3774                                  ;*SW14=1         LOOP ON TEST
3775                                  ;*SW11=1         INHIBIT ITERATIONS
3776                                  ;*SW09=1         LOOP ON ERROR
3777                                  ;*SW08=1         LOOP ON TEST IN SWR<6:0>
3778                                  ;*CALL
3779                                  ;*      SCOPE           ;;SCOPE=IOT
3780
3781  020734                          $SCOPE:
3782  020734  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
3783  020736  032777  040000  160174  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
3784  020744  001114                          BNE     $OVER           ;;YES IF SW14=1
3785                                  ;#####START OF CODE FOR THE XOR TESTER#####
3786  020746  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3787                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3788  020750  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3789  020754  012737  020774  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
3790  020762  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
3791  020766  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3792  020772  000466                          BR      $SVLAD          ;;GO TO THE NEXT TEST
3793  020774  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
3794  020776  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3795  021002  000426                          BR      7$              ;;LOOP ON THE PRESENT TEST
3796  021004                          6$:;#####END OF CODE FOR THE XOR TESTER#####
3797  021004  032777  000400  160126          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
3798  021012  001407                          BEQ     2$              ;;BR IF NO
3799  021014  017746  160120                  MOV     @SWR,-(SP)      ;;SET DESIRED TEST NUM. FROM SWR
3800  021020  042716  000200                  BIC     #$SWRMK,(SP)    ;;STRIP AWAY UNDESIRED BITS
3801  021024  122637  001102                  CMPB    (SP)+,$TSTNM    ;;ON THE RIGHT TEST?
3802  021030  001462                          BEQ     $OVER           ;;BR IF YES
3803  021032  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
3804  021036  001421                          BEQ     3$              ;;BR IF NO
3805  021040  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
3806  021046  101015                          BHI     3$              ;;BR IF NO
```

```
3807  021050  032777  001000  160062          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
3808  021056  001404                          BEQ     4$              ;;BR IF NO
3809  021060  013737  001110  001106  7$.     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
3810  021066  000443                          BR      $OVER
3811  021070  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
3812  021074  005037  001162                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3813  021100  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
3814  021102  032777  004000  160030  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
3815  021110  001013                          BNE     1$              ;;BR IF YES
3816  021112  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
3817  021116  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
3818  021120  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
3819  021124  023737  001162  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
3820  021132  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
3821  021134  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
3822  021142  013737  021212  001162          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
3823  021150  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
3824  021154  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
3825  021160  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
3826  021164  005037  001164                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3827  021170  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3828  021176  013777  001102  157736  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
3829  021204  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
3830  021210  000002                          RTI                     ;;FIXES PS
3831  021212  000764                  $MXCNT: 500.                    ;;MAX. NUMBER OF ITERATIONS
3832
3833                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
3834
3835                                  ;;**************************************************
3836                                  ;*SAVE R0-R5
3837                                  ;*CALL:
3838                                  ;*      SAVREG
3839                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
3840                                  ;*
3841                                  ;*TOP---(+16)
3842                                  ;* +2---(+18)
3843                                  ;* +4---R5
3844                                  ;* +6---R4
3845                                  ;* +8---R3
3846                                  ;*+10---R2
3847                                  ;*+12---R1
3848                                  ;*+14---R0
3849
3850  021214                          $SAVREG:
3851  021214  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3852  021216  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3853  021220  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3854  021222  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3855  021224  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
3856  021226  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
3857  021230  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
3858  021234  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
3859  021240  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
3860  021244  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
3861  021250  000002                          RTI
3862
```

```
                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
021252                                  $RESREG:
021252  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
021256  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
021262  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
021266  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
021272  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
021274  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
021276  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
021300  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
021302  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
021304  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
021306  000002                                  RTI

                                        .SBTTL  TRAP DECODER

                                        ;;*****************************************************
                                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                        ;*GO TO THAT ROUTINE.

021310  010046                          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
021312  016600  000002                          MOV     2(SP),R0        ;;GET TRAP ADDRESS
021316  005740                                  TST     -(R0)           ;;BACKUP BY 2
021320  111000                                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
021322  006300                                  ASL     R0              ;;POSITION FOR INDEXING
021324  016000  021344                          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
021330  000200                                  RTS     R0              ;;GO TO ROUTINE


                                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

021332  011646                          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
021334  016666  000004  000002                  MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
021342  000002                                  RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.

                                        ;       ROUTINE
                                        ;       -------
021344  021332                          $TRPAD: .WORD   $TRAP2
021346  017000                                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
021350  017244                                  $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
021352  017220                                  $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
021354  017260                                  $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
021356  017446                                  $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

021360  017742                                  $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

021362  017672                                  $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
```

```
3919  021364  020154                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
3920  021366  020274                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
3921  021370  020574                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
3922  021372  021214                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
3923  021374  021252                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
3924
3925                                  ;THIS ROUTINE IS USED TO ENSURE THAT THE BUS ADDRESS
3926                                  ;OF THE RP11 IS SETUP TO READ THE PROPER VALUE.
3927                                  ;IT WILL ALSO READ THE ADDRESS FROM THE TTY IF
3928                                  ;REQUIRED.
3929                                  ;NOTE: THIS ROUTINE DOES NOT PROTECT R0-R4
3930
3931  021376  005737  001204          GETADR: TST     BUSADR          ;INPUT FROM TTY REQUESTED?
3932  021402  001514                          BEQ     5$              ;NO--BRANCH
3933  021404  005037  001204                  CLR     BUSADR          ;YES--CLEAR THE REQUEST FLAG
3934  021410                          1$:
3935  021410  104401  021416                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3936  021414  000405                          BR      64$             ;;GET OVER THE ASCIZ
3937                                  ;;65$:  .ASCIZ  <15><12>/RPADR=/
3938  021430                          64$:
3939  021430  013746  001210                  MOV     RPADR,-(SP)     ;;SAVE RPADR FOR TYPEOUT
3940                                                                  ;;RP11 ADDRESS
3941  021434  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3942  021436     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
3943  021437     001                          .BYTE   1               ;;TYPE LEADING ZEROS
3944  021440  104401  021446                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3945  021444  000401                          BR      66$             ;;GET OVER THE ASCIZ
3946                                  ;;67$:  .ASCIZ  /?/
3947  021450                          66$:
3948  021450  104412                          RDOCT                   ;GET NEW RP11 ADDRESS
3949  021452  012600                          MOV     (SP)+,R0        ;SAVE THE ADDRESS
3950  021454  001402                          BEQ     2$              ;BR IF ZERO ENTRY OR A 'CR'
3951  021456  010037  001210                  MOV     R0,RPADR        ;STORE THE ADDRESS
3952  021462                          2$:
3953  021462  104401  021470                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3954  021466  000404                          BR      68$             ;;GET OVER THE ASCIZ
3955                                  ;;69$:  .ASCIZ  /RPVEC=/
3956  021500                          68$:
3957  021500  013746  001212                  MOV     RPVEC,-(SP)     ;;SAVE RPVEC FOR TYPEOUT
3958                                                                  ;;RP11 VECTOR ADDRESS
3959  021504  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3960  021506     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
3961  021507     001                          .BYTE   1               ;;TYPE LEADING ZEROS
3962  021510  104401  021516                  TYPE    ,71$            ;;TYPE ASCIZ STRING
3963  021514  000401                          BR      70$             ;;GET OVER THE ASCIZ
3964                                  ;;71$:  .ASCIZ  /?/
3965  021520                          70$:
3966  021520  104412                          RDOCT                   ;READ NEW RP11 VECTOR
3967  021522  012600                          MOV     (SP)+,R0        ;STORE THE VECTOR ADDRESS
3968  021524  001402                          BEQ     3$              ;BR IF ZERO ENTRY OR 'CR'
3969  021526  010037  001212                  MOV     R0,RPVEC        ;SAVE NEW RP11 VECTOR ADDRESS
3970  021532                          3$:
3971  021532  104401  021540                  TYPE    ,73$            ;;TYPE ASCIZ STRING
3972  021536  000404                          BR      72$             ;;GET OVER THE ASCIZ
3973                                  ;;73$:  .ASCIZ  /RPPRIO=/
3974  021550                          72$:
```

```
3975  021550  013700  001216                  MOV     RPPRIO,R0       ;CONVERT PRIORITY FOR TYPEOUT
3976  021554  006300                          ASL     R0
3977  021556  006300                          ASL     R0
3978  021560  006300                          ASL     R0
3979  021562  000300                          SWAB    R0              ;ALIGN FOR TYPEOUT
3980  021564  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
3981                                                                  ;;RP11 PRIORITY
3982  021566  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
3983  021570     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
3984  021571     001                          .BYTE   1               ;;TYPE LEADING ZEROS
3985  021572  104401  021600                  TYPE    ,75$            ;;TYPE ASCIZ STRING
3986  021576  000401                          BR      74$             ;;GET OVER THE ASCIZ
3987                                  75$:    .ASCIZ  @/@
3988  021602                          74$:
3989  021602  104412                          RDOCT                   ;GET NEW PRIORITY LEVEL
3990  021604  012600                          MOV     (SP)+,R0        ;SAVE NEW PRICRITY
3991  021606  001407                          BEQ     4$              ;BR IF ZERO ENTRY OR A 'CR'
3992  021610  006300                          ASL     R0              ;CONVERT TO PRIOIRITY VALUE
3993  021612  006300                          ASL     R0              ;CONVERT TO PRIOIRITY VALUE
3994  021614  006300                          ASL     R0              ;CONVERT TO PRIOIRITY VALUE
3995  021616  006300                          ASL     R0              ;CONVERT TO PRIOIRITY VALUE
3996  021620  006300                          ASL     R0              ;CONVERT TO PRIOIRITY VALUE
3997  021622  010037  001216                  MOV     R0,RPPRIO       ;SAVE THE VALUE
3998  021626  013777  001216  157360  4$:     MOV     RPPRIO,@RPVEC+2 ;LOAD NEW PRIORITY VALUE
3999  021634  013701  000004          5$:     MOV     ERRVEC,R1       ;SAVE THE ERROR VECTOR
4000  021640  012737  021660  000004          MOV     #6$,ERRVEC      ;SETUP FOR TRAP
4001  021646  005777  157336                  TST     @RPADR          ;CHECK FOR RP11
4002  021652  010137  000004                  MOV     R1,ERRVEC       ;RESTORE ERROR VECTOR
4003  021656  000207                          RTS     PC              ;RETURN
4004  021660  010137  000004          6$:     MOV     R1,ERRVEC       ;RESTORE ERROR VECTOR
4005  021664  022626                          CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
4006  021666  104001                          ERROR   1               ;REPORT THE ERROR
4007  021670  005737  000042                  TST     @#42            ;IS THERE A MONITOR?
4008  021674  001645                          BEQ     1$              ;NO--GO ASK FOR ADDRESS AGAIN
4009  021676  005037  016312                  CLR     %EOPCT          ;NO PASSES
4010  021702  000137  016162                  JMP     %EOP            ;GO TO END OF PROGRAM
4011
4012                                  ;ROUTINE TO CLEAR THE RP11E IN MAINTENANCE MODE
4013
4014  021706  012737  060001  001206  CLRP:   MOV     #BIT14!BIT13!BIT00,TPL
4015  021714  012764  000001  000004          MOV     #1,RPCS(R4)     ;CLEAR THE CONTROLLER
4016  021722  004737  021756                  JSR     PC,T3P          ;GENERATE 3 CLOCK PULSES
4017  021726  052764  020000  000022          BIS     #BIT13,RPM3(R4) ;SET 'SUOL'
4018  021734  000207                          RTS     PC              ;RETURN
4019
4020                                  ;GENERATE CLOCK PULSES
4021
4022  021736  012046                  TIMEP:  MOV     (R0)+,-(SP)     ;GET NUMBER OF CLOCK PULSES
4023  021740  013764  001206  000022  1$:     MOV     TPL,RPM3(R4)    ;SET SPECIFIED MAINTENANCE BITS
4024  021746  005316                          DEC     (SP)            ;DECREMENT THE COUNT
4025  021750  001373                          BNE     1$              ;BR IF MORE CLOCK PULSES
4026  021752  005726                          TST     (SP)+           ;RESTORE THE STACK POINTER
4027  021754  000200                          RTS     R0              ;RETURN
4028
4029                                  ;ROUTINE TO GENERATE 3 CLOCK PULSES
4030
```

```
4031  021756  004037  021736          T3P:    JSR     R0,TIMEP        ;GENERATE 3 CLOCK PULSES
4032  021762  000003                          .WORD   3               ;CONSTANT FOR 3 CLOCK PULSES
4033  021764  000207                          RTS     PC              ;RETURN
4034
4035                                  ;GENERATE AN INDEX PULSE
4036
4037  021766  012046                  INDEXP: MOV     (PC)+,-(SP)     ;GET THE INDEX PULSE COUNT
4038  021770  012764  030000  000022          MOV     #BIT13!BIT12,RPM3(R4)   ;GENERATE AN INDEX PULSE
4039  021776  012764  020000  000022          MOV     #BIT13,RPM3(R4)
4040  022004  005316                          DEC     (SP)            ;DECREMENT THE INDEX PULSE COUNTER
4041  022006  001367                          BNE     INDEXP          ;BR IF NOT DONE
4042  022010  005726                          TST     (SP)+           ;RESTORE THE STACK
4043  022012  000200                          RTS     R0              ;RETURN
4044
4045                                  ;ROUTINE TO SAVE THE RP11E REGISTERS FOR USE BY THE PROGRAM.
4046                                  ;       THE PROGRAM DOES NOT USE THE 'LIVE' REGISTERS FOR ERROR
4047                                  ;       DETERMINATION.
4048
4049  022014                          SAVRP:
4050  022014  016437  000000  001220          MOV     RPDS(R4),$RPDS  ;STORE RPDS
4051  022022  016437  000002  001222          MOV     RPER(R4),$RPER  ;STORE RPER
4052  022030  016437  000004  001224          MOV     RPCS(R4),$RPCS  ;STORE RPCS
4053  022036  016437  000006  001226          MOV     RPWC(R4),$RPWC  ;STORE RPWC
4054  022044  016437  000010  001230          MOV     RPBA(R4),$RPBA  ;STORE RPBA
4055  022052  016437  000012  001232          MOV     RPCA(R4),$RPCA  ;STORE RPCA
4056  022060  016437  000014  001234          MOV     RPDA(R4),$RPDA  ;STORE RPDA
4057  022066  016437  000016  001236          MOV     RPM1(R4),$RPM1  ;STORE RPM1
4058  022074  016437  000024  001240          MOV     SUCA(R4),$SUCA  ;SAVE SUCA
4059  022102  016437  000026  001242          MOV     SILO(R4),$SILO  ;SAVE SILO
4060  022110  000207                          RTS     PC              ;RETURN
4061
4062                                  ;MESSAGES USED BY THE PROGRAM
4063
4064  022112  020040     000          BLNKS2: .ASCIZ  /  /
4065
4066  022115     015  005012  047516  DRVOL:  .ASCII  <15><12><12>@NOTE:@
4067  022122  042524     072
4068  022125     015  005012  044124          .ASCII  <15><12><12>@THE MAINTENANCE SWITCH ON THE RP11E MUST BE@
4069  022132  020105  040515  047111
4070  022140  042524  040516  041516
4071  022146  020105  053523  052111
4072  022154  044103  047440  020116
4073  022162  044124  020105  050122
4074  022170  030461  020105  052515
4075  022176  052123  041040     105
4076  022203     015  051412  052105          .ASCII  <15><12>@SET TO 'ENABLE'.@
4077  022210  052040  020117  042447
4078  022216  040516  046102  023505
4079  022224     056
4080  022225     015  005012  051104          .ASCII  <15><12><12>@RIVE 0 MUST BE CYCLED DOWN OR TURNED OFF FOR THE@
4081  022232  053111  020105  020060
4082  022240  052515  052123  041040
4083  022246  020105  054503  046103
4084  022254  042105  042040  053517
4085  022262  020116  051117  052043
4086  022270  051125  042516  020104
```

```
 4087  022276  043117  020106  047506
 4088  022304  020122  044124     105
 4089  022311     015  050012  047522          .ASCII  <15><12>@PROGRAM TO FUNCTION PROPERLY.  IF DRIVE 0 IS CYCLED@
 4090  022316  051107  046501  052040
 4091  022324  020117  052506  041516
 4092  022332  044524  047117  050040
 4093  022340  047522  042520  046122
 4094  022346  027131  020040  043111
 4095  022354  042040  044522  042526
 4096  022362  030040  044440  020123
 4097  022370  054503  046103  042105
 4098  022376  005015  050125  020054          .ASCII  <15><12>@UP, STOP THE PROGRAM, CYCLE DOWN THE DRIVE, AND@
 4099  022404  052123  050117  052040
 4100  022412  042510  050040  047522
 4101  022420  051107  046501  020054
 4102  022426  054503  046103  020105
 4103  022434  047504  047127  052040
 4104  022442  042510  042040  044522
 4105  022450  042526  020054  047101
 4106  022456     104
 4107  022457     015  051012  051505          .ASCIZ  <15><12>@RESTART THE PROGRAM@<15><12><12>
 4108  022464  040524  052122  052040
 4109  022472  042510  050040  047522
 4110  022500  051107  046501  005015
 4111  022506  000012
 4112
 4113                                          .EVEN
 4114
 4115                                  .SBTTL  ERROR MESSAGES
 4116
 4117  022510  050122  030461  042040  EM1:    .ASCIZ  @RP11 DIDN'T RESPOND TO ADDRESSING@
 4118  022516  042111  023516  020124
 4119  022524  042522  050123  047117
 4120  022532  020104  047524  040440
 4121  022540  042104  042522  051523
 4122  022546  047111  000107
 4123  022552  040503  023516  020124  EM2:    .ASCIZ  @CAN'T LOAD REGISTER CORRECTLY@
 4124  022560  047514  042101  051040
 4125  022566  043505  051511  042524
 4126  022574  020122  047503  051122
 4127  022602  041505  046124  000131
 4128  022610  042522  042523  020124  EM3:    .ASCIZ  @RESET DIDN'T CLEAR REGISTER@
 4129  022616  044504  047104  052047
 4130  022624  041440  042514  051101
 4131  022632  051040  043505  051511
 4132  022640  042524  000122
 4133  022644  047503  052116  047522  EM4:    .ASCIZ  @CONTROLLER CLEAR DIDN'T CLEAR REGISTER@
 4134  022652  046114  051105  041440
 4135  022660  042514  051101  042040
 4136  022666  042111  023516  020124
 4137  022674  046103  040505  020122
 4138  022702  042522  044507  052123
 4139  022710  051105     000
 4140  022713     106  047514  052101  EM5:    .ASCIZ  @FLOATING 1'S & 0'S TEST ERROR@
 4141  022720  047111  020107  023461
 4142  022726  020123  020046  023460
```

```
 4143  022734  020123  042524  052123
 4144  022742  042440  051122  051117
 4145  022750     000
 4146  022751     122  043505  051511  EM6:    .ASCIZ  @REGISTER RAPID ACCESS TEST ERROR@
 4147  022756  042524  020122  040522
 4148  022764  044520  020104  041501
 4149  022772  042503  051523  052040
 4150  023000  051505  020124  051105
 4151  023006  047522  000122
 4152  023012  040503  023516  020124  EM7:    .ASCIZ  @CAN'T SET 'SURDY' BIT@
 4153  023020  042523  020124  051447
 4154  023026  051125  054504  020047
 4155  023034  044502  000124
 4156  023040  040503  023516  020124  EM10:   .ASCIZ  @CAN'T CLEAR THE 'SURDY' BIT@
 4157  023046  046103  040505  020122
 4158  023054  044124  020105  051447
 4159  023062  051125  054504  020047
 4160  023070  044502  000124
 4161  023074  040503  023516  020124  EM11:   .ASCIZ  @CAN'T SET 'SUSI'@
 4162  023102  042523  020124  051447
 4163  023110  051525  023511     000
 4164  023115     047  051504  020113  EM12:   .ASCIZ  @'DSK ERR' NOT SET WITH 'SUSI'@
 4165  023122  051105  023522  047040
 4166  023130  052117  051440  052105
 4167  023136  053440  052111  020110
 4168  023144  051447  051525  023511
 4169  023152     000
 4170  023153     047  051105  023522  EM13:   .ASCIZ  @'ERR' OR 'HE' NOT SET WITH 'SUSI'@
 4171  023160  047440  020122  044047
 4172  023166  023505  047040  052117
 4173  023174  051440  052105  053440
 4174  023202  052111  020110  051447
 4175  023210  051525  023511     000
 4176  023215     103  047101  052047  EM14:   .ASCIZ  @CAN'T SET 'SUFU'@
 4177  023222  051440  052105  023440
 4178  023230  052523  052506  000047
 4179  023236  040503  023516  020124  EM15:   .ASCIZ  @CAN'T CLEAR 'SUFU'@
 4180  023244  046103  040505  020122
 4181  023252  051447  043125  023525
 4182  023260     000
 4183  023261     103  047101  052047  EM16:   .ASCIZ  @CAN'T SET 'SUMP'@
 4184  023266  051440  052105  023440
 4185  023274  052523  050127  000047
 4186  023302  040503  023516  020124  EM17:   .ASCIZ  @CAN'T CLEAR 'SUMP'@
 4187  023310  046103  040505  020122
 4188  023316  051447  053525  023520
 4189  023324     000
 4190  023325     103  047101  052047  EM20:   .ASCIZ  @CAN'T SET ATTENTION BIT@
 4191  023332  051440  052105  040440
 4192  023340  052124  047105  044524
 4193  023346  047117  041040  052111
 4194  023354     000
 4195  023355     103  047101  052047  EM21:   .ASCIZ  @CAN'T CLEAR ATTENTION BIT@
 4196  023362  041440  042514  051101
 4197  023370  040440  052124  047105
 4198  023376  044524  047117  041040
```

```
4199  023404  052111  000
4200  023407  122  051505  052105  EM22:  .ASCIZ  @RESET DIDN'T CLEAR ATTENTION BITS@
4201  023414  042040  042111  023516
4202  023422  020124  046103  040505
4203  023430  020122  052101  042524
4204  023436  052116  047511  020116
4205  023444  044502  051524  000
4206  023451  101  052124  047105  EM23:  .ASCIZ  @ATTENTION BITS CLEARED BY WRITING 0'S INTO RPDS@
4207  023456  044524  047117  041040
4208  023464  052111  020123  046103
4209  023472  040505  042522  020104
4210  023500  054502  053440  044522
4211  023506  044524  043516  030040
4212  023514  051447  044440  052116
4213  023522  020117  050122  051504
4214  023530  000
4215  023531  103  047101  052047  EM24:  .ASCIZ  @CAN'T SET 'WPV' ERROR@
4216  023536  051440  052105  023440
4217  023544  050127  023526  042440
4218  023552  051122  051117  000
4219  023557  047  051105  023522  EM25:  .ASCIZ  @'ERR' OR 'ME' NOT SET WITH 'WPV'@
4220  023564  047440  020122  044047
4221  023572  023505  047040  052117
4222  023600  051440  052105  053440
4223  023606  052111  020110  053447
4224  023614  053120  000047
4225  023620  040503  023516  020124  EM26:  .ASCIZ  @CAN'T SET 'FUV' ERROR@
4226  023626  042523  020124  043047
4227  023634  053125  020047  051105
4228  023642  047522  000122
4229  023646  042447  051122  020047  EM27:  .ASCIZ  @'ERR' OR 'ME' NOT SET WITH 'FUV'@
4230  023654  051117  023440  042510
4231  023662  020047  047516  020124
4232  023670  042523  020124  044527
4233  023676  044124  023440  052506
4234  023704  023526  000
4235  023707  047  054116  023503  EM30:  .ASCIZ  @'NXC' ERROR SET WITH VALID CYLINDER ADDRESS@
4236  023714  042440  051122  051117
4237  023722  051440  052105  053440
4238  023730  052111  020110  040526
4239  023736  044514  020104  054503
4240  023744  044514  042116  051105
4241  023752  040440  042104  042522
4242  023760  051523  000
4243  023763  047  054116  023503  EM31:  .ASCIZ  @'NXC' DIDN'T SET WITH INVALID CYLINDER ADDRESS@
4244  023770  042040  042111  023516
4245  023776  020124  042523  020124
4246  024004  044527  044124  044440
4247  024012  053116  046101  042111
4248  024020  041440  046131  047111
4249  024026  042504  020122  042101
4250  024034  051104  051505  000123
4251  024042  042447  051122  020047  EM32:  .ASCIZ  @'ERR' OR 'ME' DIDN'T SET WITH 'NXC'@
4252  024050  051117  023440  042510
4253  024056  020047  044504  047104
4254  024064  052047  051440  052105
```

```
4255  024072  053440  052111  020110
4256  024100  047047  041530  000047
4257  024106  047047  052130  020047  EM33:   .ASCIZ  @'NXT' ERROR SET WITH VALID TRACK ADDRESS@
4258  024114  051105  047522  020122
4259  024122  042523  020124  044527
4260  024130  044124  053040  046101
4261  024136  042111  052040  040522
4262  024144  045503  040440  042104
4263  024152  042522  051523     000
4264  024157     047  054116  023524  EM34:   .ASCIZ  @'NXT' DIDN'T SET WITH INVALID TRACK ADDRESS@
4265  024164  042040  042111  023516
4266  024172  020124  042523  020124
4267  024200  044527  044124  044440
4268  024206  053116  046101  042111
4269  024214  052040  040522  045503
4270  024222  040440  042104  042522
4271  024230  051523     000
4272  024233     047  051105  023522  EM35:   .ASCIZ  @'ERR' OR 'ME' DIDN'T SET WITH 'NXT'@
4273  024240  047440  020122  044047
4274  024246  023505  042040  042111
4275  024254  023516  020124  042523
4276  024262  020124  044527  044124
4277  024270  023440  054116  023524
4278  024276     000
4279  024277     047  054116  023523  EM36:   .ASCIZ  @'NXS' ERROR SET WITH VALID SECTOR ADDRESS@
4280  024304  042440  051122  051117
4281  024312  051440  052105  053440
4282  024320  052111  020110  040526
4283  024326  044514  020104  042523
4284  024334  052103  051117  040440
4285  024342  042104  042522  051523
4286  024350     000
4287  024351     047  054116  023523  EM37:   .ASCIZ  @'NXS' DIDN'T SET WITH INVALID SECTOR ADDRESS@
4288  024356  042040  042111  023516
4289  024364  020124  042523  020124
4290  024372  044527  044124  044440
4291  024400  053116  046101  042111
4292  024406  051440  041505  047524
4293  024414  020122  042101  051104
4294  024422  051505  000123
4295  024426  042447  051122  020047  EM40:   .ASCIZ  @'ERR' OR 'ME' DIDN'T SET WITH 'NXS'@
4296  024434  051117  023440  042510
4297  024442  020047  044504  047104
4298  024450  052047  051440  052105
4299  024456  053440  052111  020110
4300  024464  047047  051530  000047
4301  024472  040503  023516  020124  EM41:   .ASCIZ  @CAN'T SET 'PROG' ERROR@
4302  024500  042523  020124  050047
4303  024506  047522  023507  042440
4304  024514  051122  051117     000
4305  024521     047  051105  023522  EM42:   .ASCIZ  @'ERR' OR 'ME' NOT SET WITH 'PROG'@
4306  024526  047440  020122  044047
4307  024534  023505  047040  052117
4308  024542  051440  052105  053440
4309  024550  052111  020110  050047
4310  024556  047522  023507     000
```

```
4311  024563     103  047101  052047  EM43:   .ASCIZ  @CAN'T SET 'MODE' ERROR@
4312  024570  051440  052105  023440
4313  024576  047515  042504  020047
4314  024604  051105  047522  000122
4315  024612  042447  051122  020047  EM44:   .ASCIZ  @'ERR' OR 'HE' NOT SET WITH 'MODE'@
4316  024620  051117  023440  042510
4317  024626  020047  047516  020124
4318  024634  042523  020124  044527
4319  024642  044124  023440  047515
4320  024650  042504  000047
4321  024654  040503  023516  020124  EM45:   .ASCIZ  @CAN'T CLEAR 'SUSI'@
4322  024662  046103  040505  020122
4323  024670  051447  051525  023511
4324  024676     000
4325  024677     116  020117  052101  EM46:   .ASCIZ  @NO ATTENTION INTERRUPT@
4326  024704  042524  052116  047511
4327  024712  020116  047111  042524
4328  024720  051122  050125  000124
4329  024726  040447  042511  020047  EM47:   .ASCIZ  @'AIE' DIDN'T CLEAR WHEN INTERRUPT OCCURED@
4330  024734  044504  047104  052047
4331  024742  041440  042514  051101
4332  024750  053440  042510  020116
4333  024756  047111  042524  051122
4334  024764  050125  020124  041517
4335  024772  052503  042522  000104
4336  025000  047516  044440  052116  EM50:   .ASCIZ  @NO INTERRUPT WITH ATTENTION BITS 0 & 1 SET@
4337  025006  051105  052522  052120
4338  025014  053440  052111  020110
4339  025022  052101  042524  052116
4340  025030  047511  020116  044502
4341  025036  051524  030040  023040
4342  025044  030440  051440  052105
4343  025052     000
4344  025053     123  041505  047117  EM51:   .ASCIZ  @SECOND INTERRUPT DIDN'T OCCUR WITH ATTENTION BIT 1 SET@
4345  025060  020104  047111  042524
4346  025066  051122  050125  020124
4347  025074  044504  047104  052047
4348  025102  047440  041503  051125
4349  025110  053440  052111  020110
4350  025116  052101  042524  052116
4351  025124  047511  020116  044502
4352  025132  020124  020061  042523
4353  025140  000124
4354  025142  047516  044440  052116  EM52:   .ASCIZ  @NO INTERRUPT FROM CONTROLLER 'READY'@
4355  025150  051105  052522  052120
4356  025156  043040  047522  020115
4357  025164  047503  052116  047522
4358  025172  046114  051105  023440
4359  025200  042522  042101  023531
4360  025206     000
4361  025207     047  042522  042101  EM53:   .ASCIZ  @'READY' INTERRUPT WITH 'IDE' SET@
4362  025214  023531  044440  052116
4363  025222  051105  052522  052120
4364  025230  053440  052111  020110
4365  025236  044447  042504  020047
4366  025244  042523  000124
```

```
 4367  025250  047516  044440  052116  EM54:   .ASCIZ  @NO INTERRUPT FROM RP11 WITH CPU AT PR4@
 4368  025256  051105  052522  052120
 4369  025264  043040  047522  020115
 4370  025272  050122  030461  053440
 4371  025300  052111  020110  050103
 4372  025306  020125  05\101  050040
 4373  025314  032122     000
 4374  025317     111  052116  051105  EM55:   .ASCIZ  @INTERRUPT FROM RP11 WITH CPU AT PR5@
 4375  025324  052522  052120  043040
 4376  025332  047522  020115  050122
 4377  025340  030461  053440  052111
 4378  025346  020110  050103  020125
 4379  025354  052101  050040  032522
 4380  025362     000
 4381  025363     111  052116  051105  EM56:   .ASCIZ  @INTERRUPT FROM RP11 WITH CPU AT PR6@
 4382  025370  052522  052120  043040
 4383  025376  047522  020115  050122
 4384  025404  030461  053440  052111
 4385  025412  020110  050103  020125
 4386  025420  052101  050040  033122
 4387  025426     000
```

```
 4388  025427     111  052116  051105  EM57:   .ASCIZ  @INTERRUPT FROM RP11 WITH CPU AT PR7@
 4389  025434  052522  052120  043040
 4390  025442  047522  020115  050122
 4391  025450  030461  053440  052111
 4392  025456  020110  050103  020125
 4393  025464  052101  050040  033522
 4394  025472     000
 4395  025473     047  042522  042101  EM60:   .ASCIZ  @'READY' DIDN'T CLEAR AT END OF OPERATION@
 4396  025500  023531  042040  042111
 4397  025506  023516  020124  046103
 4398  025514  040505  020122  052101
 4399  025522  042440  042116  047440
 4400  025530  020106  050117  051105
 4401  025536  052101  047511  000116
 4402  025544  051447  041525  023501  EM61:   .ASCIZ  @'SUCA' INCORRECT@
 4403  025552  044440  041516  051117
 4404  025560  042522  052103     000
 4405  025566     111  042116  054105  EM62:   .ASCIZ  @INDEX DIDN'T CLEAR 'SOT'@
 4406  025572  042040  042111  023516
 4407  025600  020124  046103  040505
 4408  025606  020122  051447  052117
 4409  025614  000047
 4410  025616  051447  052117  020047  EM63:   .ASCIZ  @'SOT' DIDN'T COUNT CORRECTLY@
 4411  025624  044504  047104  052047
 4412  025632  041440  052517  052116
 4413  025640  041440  051117  042522
 4414  025646  052103  054514     000
 4415  025653     103  047117  051124  EM64:   .ASCIZ  @CONTROLLER CLEAR DIDN'T SET SILO INPUT READY@
 4416  025660  046117  042514  020122
 4417  025666  046103  040505  020122
 4418  025674  044504  047104  052047
 4419  025702  051440  052105  051440
 4420  025710  046111  020117  047111
 4421  025716  052520  020124  042522
 4422  025724  042101  000131
 4423  025730  044523  047514  047440  EM65:   .ASCIZ  @SILO OUTPUT READY NOT SET AFTER WORD LOADED INTO SILO@
 4424  025736  052125  052520  020124
 4425  025744  042522  042101  020131
 4426  025752  047516  020124  042523
 4427  025760  020124  043101  042524
 4428  025766  020122  047527  042122
 4429  025774  046040  040517  042504
 4430  026002  020104  047111  047524
 4431  026010  051440  046111  000117
 4432  026016  040504  040524  051040  EM66:   .ASCIZ  @DATA READ FROM SILO IS INCORRECT@
 4433  026024  040505  020104  051106
 4434  026032  046517  051440  046111
 4435  026040  020117  051511  044440
 4436  026046  041516  051117  042522
 4437  026054  052103     000
 4438  026057     123  046111  020117  EM67:   .ASCII  @SILO INPUT READY DIDN'T CLEAR AFTER 64 WORDS WERE@
 4439  026064  047111  052520  020124
 4440  026072  042522  042101  020131
 4441  026100  044504  047104  052047
 4442  026106  041440  042514  051101
 4443  026114  040440  052106  051105
```

```
 4444  026122  033040  020064  047527
 4445  026130  042122  020123  042527
 4446  026136  042522
 4447  026140  005015  047514  042101          .ASCIZ  <15><12>@LOADED INTO THE SILO@
 4448  026146  042105  044440  052116
 4449  026154  020117  044124  020105
 4450  026162  044523  047514     000
 4451  026167     123  046111  020117  EM70:   .ASCIZ  @SILO OUTPUT READY DIDN'T CLEAR AFTER SILO EMPTIED@
 4452  026174  052517  050124  052135
 4453  026202  051040  040505  054504
 4454  026210  042040  042111  023516
 4455  026216  020124  046103  040505
 4456  026224  020122  043101  042524
 4457  026232  020122  044523  047514
 4458  026240  042440  050115  044524
 4459  026246  042105     000
 4460  026251     102  051525  047440  EM71:   .ASCIZ  @BUS OUT LINES TO THE DRIVE WERE NOT CLEARED BY CONTROLLER CLEAR@
 4461  026256  052125  046040  047111
 4462  026264  051505  052040  020117
 4463  026272  044124  020105  051104
 4464  026300  053111  020105  042527
 4465  026306  042522  047040  052117
 4466  026314  041440  042514  051101
 4467  026322  042105  041040  020131
 4468  026330  047503  052116  047522
 4469  026336  046114  051105  041440
 4470  026344  042514  051101     000
 4471  026351     104  044522  042526  EM72:   .ASCII  @DRIVE BUS ERROR; 'SET CYLINDER' AND CYLINDER ADDRESS@
 4472  026356  041040  051525  042440
 4473  026364  051122  051117  020073
 4474  026372  051447  052105  041440
 4475  026400  046131  047111  042504
 4476  026406  023522  040440  042116
 4477  026414  041440  046131  047111
 4478  026422  042504  020122  042101
 4479  026430  051104  051505     123
 4480  026435     015  051412  047510          .ASCIZ  <15><12>@SHOULD BE ON THE BUS@
 4481  026442  046125  020104  042502
 4482  026450  047440  020116  044124
 4483  026456  020105  052502  000123
 4484  026464  051104  053111  020105  EM73:   .ASCIZ  @DRIVE BUS ERROR: ONLY 'RESET HEAD' SHOULD BE ON THE BUS@
 4485  026472  052502  020123  051105
 4486  026500  047522  035122  047440
 4487  026506  046116  020131  051047
 4488  026514  051505  052105  044040
 4489  026522  040505  023504  051440
 4490  026530  047510  046125  020104
 4491  026536  042502  047440  020116
 4492  026544  044124  020105  052502
 4493  026552  000123
 4494  026554  051104  053111  020105  EM74:   .ASCIZ  @DRIVE BUS ERROR: 'SET HEAD' & HEAD ADDRESS SHOULD BE ON THE BUS@
 4495  026562  052502  020123  051105
 4496  026570  047522  035122  023440
 4497  026576  042523  020124  042510
 4498  026604  042101  020047  020046
 4499  026612  042510  042101  040440
```

```
4500  026620  042104  042522  051523
4501  026626  051440  047510  046125
4502  026634  020104  042502  047440
4503  026642  020116  044124  020105
4504  026650  052502  000123
4505  026654  051104  053111  020105  EM75:   .ASCIZ  @DRIVE BUS ERROR: ONLY 'SEEK START' SHOULD BE ON THE BUS@
4506  026662  052502  020123  051105
4507  026670  047522  035122  047440
4508  026676  046116  020131  051447
4509  026704  042505  020113  052123
4510  026712  051101  023524  051440
4511  026720  047510  046125  020104
4512  026726  042502  047440  020116
4513  026734  044124  020105  052502
4514  026742  000123
4515  026744  051104  053111  020105  EM76:   .ASCIZ  @DRIVE BUS ERROR: 'RESTORE' AND 'CONTROL' SHOULD BE ON THE BUS@
4516  026752  052502  020123  051105
4517  026760  047522  035122  023440
4518  026766  042522  052123  051117
4519  026774  023505  040440  042116
4520  027002  023440  047503  052116
4521  027010  047522  023514  051440
4522  027016  047510  046125  020104
4523  027024  042502  047440  020116
4524  027032  044124  020105  052502
4525  027040  000123
4526  027042  051104  053111  020105  EM77:   .ASCII  @DRIVE BUS ERROR: 'CONTROL', 'SELECT HEAD', AND 'READ'@
4527  027050  052502  020123  051105
4528  027056  047522  035122  023440
4529  027064  047503  052116  047522
4530  027072  023514  020054  051447
4531  027100  046105  041505  020124
4532  027106  042510  042101  026047
4533  027114  040440  042116  023440
4534  027122  042522  042101     047
4535  027127     015  051412  047510          .ASCIZ  <15><12>@SHOULD BE ON THE BUS@
4536  027134  046125  020104  042502
4537  027142  047440  020116  044124
4538  027150  020105  052502  000123
4539  027156  051104  053111  020105  EM100:  .ASCII  @DRIVE BUS ERROR: 'CONTROL', SELECT HEAD', 'ERASE',@
4540  027164  052502  020123  051105
4541  027172  047522  035122  023440
4542  027200  047503  052116  047522
4543  027206  023514  020054  042523
4544  027214  042514  052103  044040
4545  027222  040505  023504  020054
4546  027230  042447  040522  042523
4547  027236  026047
4548  027240  005015  047101  020104          .ASCIZ  <15><12>@AND 'WRITE' SHOULD BE ON THE BUS@
4549  027246  053447  044522  042524
4550  027254  020047  044123  052517
4551  027262  042114  041040  020105
4552  027270  047117  052040  042510
4553  027276  041040  051525     000
4554  027303     047  042522  042101  EM101:  .ASCIZ  @'READY' DIDN'T CLEAR WHEN 'GO' WAS SET@
4555  027310  023531  042040  042111
```

```
4556  027316  023516  020124  046103
4557  027324  040505  020122  044127
4558  027332  047105  023440  047507
4559  027340  020047  040527  020123
4560  027346  042523  000124
4561  027352  040503  023516  020124  EM102:  .ASCIZ  @CAN'T SET 'SUOL'@
4562  027360  042523  020124  051447
4563  027366  047525  023514      000
4564  027373      103  047101  052047  EM103:  .ASCIZ  @CAN'T CLEAR 'SUOL'@
4565  027400  041440  042514  051101
4566  027406  023440  052523  046117
4567  027414  000047
4568  027416  040503  023516  020124  EM104:  .ASCIZ  @CAN'T SET 'SUSU'@
4569  027424  042523  020124  051447
4570  027432  051525  023525      000
4571  027437      103  047101  052047  EM105:  .ASCIZ  @CAN'T CLEAR 'SUSU'@
4572  027444  041440  042514  051101
4573  027452  023440  052523  052123
4574  027460  020047
4575  027462  044447  042504  020047  EM106:  .ASCIZ  @'IDE' NOT SET AFTER INTERRUPT@
4576  027470  047516  020124  042523
4577  027476  020124  043101  042124
4578  027504  020122  047111  042124
4579  027512  051122  050125  000124
4580  027520  052101  047124  044440  EM107:  .ASCIZ  @ATTN INTERRUPT OCCURED WITH NO ATTN BITS SET@
4581  027526  052116  051105  052122
4582  027534  052120  047440  041503
4583  027542  051125  042105  053440
4584  027550  052111  020110  047516
4585  027556  040440  052124  020116
4586  027564  044502  051524  051440
4587  027572  052105      000
4588  027575      123  046111  020117  EM110:  .ASCIZ  @SILO NOT FULL, 'INPUT READY' NOT SET@
4589  027602  047516  020124  052506
4590  027610  046114  020054  044447
4591  027616  050116  052125  051040
4592  027624  043505  054504  020047
4593  027632  047516  020124  042523
4594  027640  000124
4595  027642  044523  047514  047040  EM111:  .ASCIZ  @SILO NOT EMPTY, 'OUTPUT READY' NOT SET@
4596  027650  052117  042440  050115
4597  027656  054524  020054  047447
4598  027664  052125  052520  020124
4599  027672  042522  042101  023531
4600  027700  047040  052117  051440
4601  027706  052105      000
4602  027711      047  047111  052111  EM112:  .ASCIZ  @'INIT' COMMAND SET 'PROG' ERROR@
4603  027716  020047  047503  046515
4604  027724  047101  020104  042523
4605  027732  020124  050047  047522
4606  027740  023507  042440  051122
4607  027746  051117      000
4608
4609                                  ;ERROR HEADER (DH) MESSAGES
4610
4611  027751      105  051122  050040  DH1:    .ASCIZ  @ERR PC  RPADR@
```

```
 4612  027756  020103  051040  040520
 4613  027764  051104     000
 4614  027767     124  051505  020124  DH2:    .ASCIZ  @TEST # ERR PC  REG ADR EXPTD   RECVD@
 4615  027774  020043  042440  051122
 4616  030002  050040  020103  051040
 4617  030010  043505  040440  051104
 4618  030016  042440  050130  042124
 4619  030024  020040  051040  041505
 4620  030032  042126     000
 4621  030035     124  051505  020124  DH7:    .ASCIZ  @TEST # ERR PC  RPDS@
 4622  030042  020043  042440  051122
 4623  030050  050040  020103  051040
 4624  030056  042120  000123
 4625  030062  042524  052123  021440  DH11:   .ASCIZ  @TEST # ERR PC  RPDS    RPER    RPCS@
 4626  030070  020040  051105  020122
 4627  030076  041520  020040  050122
 4628  030104  051504  020040  020040
 4629  030112  050122  051105  020040
 4630  030120  020040  050122  051503
 4631  030126     000
 4632  030127     124  051505  020124  DH20:   .ASCIZ  @TEST # ERR PC ATTN BIT RPDS@
 4633  030134  020043  042440  051122
 4634  030142  050040  020103  052101
 4635  030150  047124  041040  052111
 4636  030156  051040  042120  000123
 4637  030164  042524  052123  021440  DH24:   .ASCIZ  @TEST # ERR PC  RPER    RPCS@
 4638  030172  020040  051105  020122
 4639  030200  041520  020040  050122
 4640  030206  051105  020040  020040
 4641  030214  050122  051503     000
 4642  030221     124  051505  020124  DH30:   .ASCIZ  @TEST # ERR PC  RPER    RPCS    RPCA    RPCA LOADED WITH@
 4643  030226  020043  042440  051122
 4644  030234  050040  020103  051040
 4645  030242  042520  020122  020040
 4646  030250  051040  041520  020123
 4647  030256  020040  051040  041520
 4648  030264  020101  020040  051040
 4649  030272  041520  020101  047514
 4650  030300  042101  042105  053440
 4651  030306  052111  000110
 4652  030312  042524  052123  021440  DH33:   .ASCIZ  @TEST # ERR PC  RPER    RPCS    RPDA    RPDA LOADED WITH@
 4653  030320  020040  051105  020122
 4654  030326  041520  020040  050122
 4655  030334  051105  020040  020040
 4656  030342  050122  051503  020040
 4657  030350  020040  050122  040504
 4658  030356  020040  020040  050122
 4659  030364  042504  046040  040517
 4660  030372  042504  020104  044527
 4661  030400  044124     000
 4662  030403     124  051505  020124  DH46:   .ASCIZ  @TEST # ERR PC  RPDS    RPCS    LOADED ATTN BIT@
 4663  030410  020043  042440  051122
 4664  030416  050040  020103  051040
 4665  030424  042120  020123  020040
 4666  030432  051040  041520  020123
 4667  030440  020040  046040  040517
```

```
4668  030446  042504  020104  052101
4669  030454  047124  041040  052111
4670  030462     000
4671  030463     124  051505  020124  DH50:   .ASCIZ  @TEST # ERR PC  RPDS    FPCS@
4672  030470  020043  042440  051122
4673  030476  050040  020103  051040
4674  030504  042120  020123  020040
4675  030512  051040  041520  000123
4676  030520  042524  052123  021440  DH52:   .ASCIZ  @TEST # ERR PC  RPCS@ .
4677  030526  020040  051105  020122
4678  030534  041520  020040  050122
4679  030542  051503     000
4680  030545     124  051505  020124  DH61:   .ASCIZ  @TEST # ERR PC  SUCA SUCA LOADED WITH@
4681  030552  020043  042440  051122
4682  030560  050040  020103  051440
4683  030566  041525  020101  052523
4684  030574  040503  046040  040517
4685  030602  042504  020104  044527
4686  030610  044124     000
4687  030613     124  051505  020124  DH62:   .ASCIZ  @TEST # ERR PC  RPDA@
4688  030620  020043  042440  051122
4689  030626  050040  020103  051040
4690  030634  042120  000101
4691  030640  042524  052123  021440  DH63:   .ASCIZ  @TEST # ERR PC  RPDA 'SOT' SHOULD BE@
4692  030646  020040  051105  020122
4693  030654  041520  020040  050122
4694  030662  040504  023440  047523
4695  030670  023524  051440  047510
4696  030676  046125  020104  042502
4697  030704     000
4698  030705     124  051505  020124  DH64:   .ASCIZ  @TEST # ERR PC  RPM1  SILO REGISTER@
4699  030712  020043  042440  051122
4700  030720  050040  020103  051040
4701  030726  046520  020061  051440
4702  030734  046111  020117  042522
4703  030742  044507  052123  051105
4704  030750     000
4705  030751     124  051505  020124  DH65:   .ASCIZ  @TEST # ERR PC  RPM1@
4706  030756  020043  042440  051122
4707  030764  050040  020103  051040
4708  030772  046520  000061
4709  030776  042524  052123  021440  DH66:   .ASCIZ  @TEST # ERR PC  RPM1    SILO    EXPTD   RECVD@
4710  031004  020040  051105  020122
4711  031012  041520  020040  050122
4712  031020  030515  020040  020040
4713  031026  044523  047514  020040
4714  031034  020040  054105  052120
4715  031042  020104  020040  042522
4716  031050  053103  000104
4717  031054  042524  052123  021440  DH71:   .ASCIZ  @TEST # ERR PC  RPDS    RPCS    FPCA    RPDA    RPM1@
4718  031062  020040  051105  020122
4719  031070  041520  020040  050122
4720  031076  051504  020040  020040
4721  031104  050122  051503  020040
4722  031112  020040  050122  040503
4723  031120  020040  020040  050122
```

```
 4724  031126  040504  020040  020040
 4725  031134  050122  030515     000
 4726  031141     124  051505  020124  DH72:   .ASCIZ  @TEST # ERR PC  RPDS    RPCS    RPCA    RPM1@
 4727  031146  020043  042440  051122
 4728  031154  050040  020103  051040
 4729  031162  042120  020123  020040
 4730  031170  051040  041520  020123
 4731  031176  020040  051040  041520
 4732  031204  020101  020040  051040
 4733  031212  046520  000061
 4734  031216  042524  052123  021440  DH73:   .ASCIZ  @TEST # ERR PC  RPDS    RPCS    RPM1 EXPTD BUS@
 4735  031224  020040  051105  020122
 4736  031232  041520  020040  050122
 4737  031240  051504  020040  020040
 4738  031246  050122  051503  020040
 4739  031254  020040  050122  030515
 4740  031262  042440  050130  042124
 4741  031270  041040  051525     000
 4742  031275     124  051505  020124  DH74:   .ASCIZ  @TEST # ERR PC  RPDS    RPCS    RPDA    RPM1@
 4743  031302  020043  042440  051122
 4744  031310  050040  020103  051040
 4745  031316  042120  020123  020040
 4746  031324  051040  041520  020123
 4747  031332  020040  051040  042120
 4748  031340  020101  020040  051040
 4749  031346  046520  000061
 4750
 4751                          .EVEN
 4752
 4753                          ;ERROR MESSAGE 'DT' ENTRIES
 4754
 4755  031352  001116  001210        DT1:    .WORD   $ERRPC,RPADR
 4756  031356  001160  001116  001122  DT2:    .WORD   $TMPO,$ERRPC,$BDADR,$GDDAT,$BDDAT
 4757  031364  001124  001126
 4758  031370  001160  001116  001220  DT7:    .WORD   $TMPO,$ERRPC,$RPDS
 4759  031376  001160  001116  001220  DT11:   .WORD   $TMPO,$ERRPC,$RPDS,$RPER,$RPCS
 4760  031404  001222  001224
 4761  031410  001160  001116  001124  DT20:   .WORD   $TMPO,$ERRPC,$GDDAT,$RPDS
 4762  031416  001220
 4763  031420  001160  001116  001222  DT24:   .WORD   $TMPO,$ERRPC,$RPER,$RPCS
 4764  031426  001224
 4765  031430  001160  001116  001222  DT30:   .WORD   $TMPO,$ERRPC,$RPER,$RPCS,$RPCA,$GDDAT
 4766  031436  001224  001232  001124
 4767  031444  001160  001116  001222  DT33:   .WORD   $TMPO,$ERRPC,$RPER,$RPCS,$RPDA,$GDDAT
 4768  031452  001224  001234  001124
 4769  031460  001160  001116  001220  DT46:   .WORD   $TMPO,$ERRPC,$RPDS,$RPCS,$GDDAT
 4770  031466  001224  001124
 4771  031472  001160  001116  001220  DT50:   .WORD   $TMPO,$ERRPC,$RPDS,$RPCS
 4772  031500  001224
 4773  031502  001160  001116  001224  DT52:   .WORD   $TMPO,$ERRPC,$RPCS
 4774  031510  001160  001116  001240  DT61:   .WORD   $TMPO,$ERRPC,$SUCA,$GDDAT
 4775  031516  001124
 4776  031520  001160  001116  001126  DT62:   .WORD   $TMPO,$ERRPC,$BDDAT
 4777  031526  001160  001116  001126  DT63:   .WORD   $TMPO,$ERRPC,$BDDAT,$GDDAT
 4778  031534  001124
 4779  031536  001160  001116  001236  DT64:   .WORD   $TMPO,$ERRPC,$RPM1,$SILO
```

```
 4780  031544  001242
 4781  031546  001160  001116  001236  DT65:   .WORD   $TMPO,$ERRPC,$RPM1
 4782  031554  001160  001116  001236  DT66:   .WORD   $TMPO,$ERRPC,$RPM1,$SILO,$GDDAT,$BDDAT
 4783  031562  001242  001124  001126
 4784  031570  001160  001116  001220  DT71:   .WORD   $TMPO,$ERRPC,$RPDS,$RPCS,$RPCA,$RPDA,$RPM1
 4785  031576  001224  001232  001234
 4786  031604  001236
 4787  031606  001160  001116  001220  DT72:   .WORD   $TMPO,$ERRPC,$RPDS,$RPCS,$RPCA,$RPM1
 4788  031614  001224  001232  001236
 4789  031622  001160  001116  001220  DT73:   .WORD   $TMPO,$ERRPC,$RPDS,$RPCS,$RPM1,$GDDAT
 4790  031630  001224  001236  001124
 4791  031636  001160  001116  001220  DT74:   .WORD   $TMPO,$ERRPC,$RPDS,$RPCS,$RPDA,$RPM1
 4792  031644  001224  001234  001236
 4793
 4794                                  ;ERROR MESSAGE 'DF' ENTRIES
 4795
 4796  031652  000001                  DF:     .WORD   1
 4797  031654     002                          .BYTE   2
 4798  031655     000                          .BYTE   0
 4799
 4800  031656  000001                  DF1:    .WORD   1
 4801  031660     003                          .BYTE   3
 4802  031661     000                          .BYTE   0
 4803
 4804  031662  000001                  DF2:    .WORD   1
 4805  031664     005                          .BYTE   5
 4806  031665     000                          .BYTE   0
 4807
 4808  031666  000001                  DF20:   .WORD   1
 4809  031670     004                          .BYTE   4
 4810  031671     000                          .BYTE   0
 4811
 4812  031672  000001                  DF30:   .WORD   1
 4813  031674     006                          .BYTE   6
 4814  031675     000                          .BYTE   0
 4815
 4816  031676  000001                  DF71:   .WORD   1
 4817  031700     007                          .BYTE   7
 4818  031701     000                          .BYTE   0
 4819
 4820          000001                       .END
```

```
BABT    006060          1830    1840#   1853    1855
BIT0  = 000001           702#
BIT00 = 000001           692#    702     1656    2039    2190    2217    2246    2273    2306    2333    2366    2392    2423
                        2452    2559    2724    2996    3075    3106    3109    3141    3154    4014
BIT01 = 000002           691#    701     3115
BIT02 = 000004           690#    700     3040
BIT03 = 000010           689#    699     3016
BIT04 = 000020           688#    698     3154
BIT05 = 000040           687#    697     3115    3154
BIT06 = 000100           686#    696     2582    2591    2631    2653    2676    2699    3081
BIT07 = 000200           685#    695
BIT08 = 000400           684#    694     2117    2123    2156    2455    2497    2751    2764    2780    2800    2802    2804
                        2806    2808    2810    2812    2814    2816    2818    2820    2822    2824    2826    2830    3109
                        3146    3148    3797
BIT09 = 001000           683#    693     2093    2099    3254    3807
BIT1  = 000002           701#
BIT10 = 002000           682#    1660    2033    2048    2069    2075    2426    3016    3040    3081    3115    3154    3238
BIT11 = 004000           681#    2035    2050    2091    2097    2217    2369    2395    2862    2915    2925    2947    3814
BIT12 = 010000           680#    2309    2336    2872    2930    2950    2955    4038
BIT13 = 020000           679#    1413    1565    1571    1577    1656    1707    2067    2190    2217    2246    2249    2273
                        3276    2306    2333    2366    2392    2421    2423    2435    2452    2482    2492    2516    2544
                        2558    2724    2764    2765    2780    2781    2800    2801    2802    2803    2804    2805    2806
                        2807    2808    2809    2810    2811    2812    2813    2814    2815    2816    2817    2818    2819
                        2820    2821    2822    2823    2824    2825    2826    2827    2830    2831    2996    3075    3106
                        3109    3141    3146    3147    3148    3245    4014    4017    4038    4039
BIT14 = 040000           678#    1567    1573    1656    2008    2014    2045    2066    2073    2190    2199    2217    2220
                        2226    2246    2273    2283    2306    2333    2343    2366    2392    2402    2423    2432    2452
                        2461    2724    2996    3075    3106    3109    3141    3783    4014
BIT15 = 100000           677#    2010    2016    2045    2115    2121    2187    2190    2193    2139    2226    2283    2343
                        2402    2432    2461    2889
BIT2  = 000004           700#
BIT3  = 000010           699#
BIT4  = 000020           698#
BIT5  = 000040           697#
BIT6  = 000100           696#
BIT7  = 000200           695#
BIT8  = 000400           694#
BIT9  = 001000           693#
BLNKS2  022112          3307    3314    3321    4064#
BPTVEC= 000014           709#
BUSADR  001204           782#    1345#   1347#   3931    3933#
CABT    006244          1866    1876#   1889    1891
CAF1    013416          2739    2743#   2752
CKSWR = 104407          3234    3253    3782    3918#
CLEARP  004252          1589    1590    1593#
CLRP    021706          1564    1727    1763    1799    1835    1871    1907    1947    1980    2007    2032    2065    2090
                        2114    2141    2186    2214    2242    2269    2302    2329    2362    2388    2420    2449    2477
                        2480    2512    2515    2542    2578    2606    2629    2651    2674    2697    2718    2743    2769
                        2799    2840    2913    2979    3065    3103    3137    3161    4014#
CR    = 000015           617#    3387    3397
CRLF  = 000200           618#    3358    3397
CSBT    005510          1758    1768#   1781    1783
CSF1    011760          2473    2477#   2498
CSF10   013112          2670    2671    2673#
CSF11   013214          2693    2694    2696#
CSF12   013302          2714    2715    2717#
```

```
CSF2    012464          2574    2575    2577#
CSF3    012610          2602    2603    2605#
CSF6    012710          2625    2626    2628#
CSF7    013010          2647    2648    2650#
DABT    006430          1902    1912#   1925    1927
DAF1    013560          2762    2771#   2784    2786
DAF2    013710          2795    2796    2798#
DDISP = 177570           624#     762    1374
DF      031652           824    4796#
DF1     031656           866     873     943     950    1111    1118    1125    1132    1139    1146    1153    1167    1188
                        1203    1210    1277    1284    1291    1298    1305    1312    1326    1333    4800#
DF2     031662           831     838     845     852     859     880     887     894     901     908     915     922    1076
                        1083    1090    1319    4804#
DF20    031666           929     936     957     964     971     978    1048    1055    1062    1069    1097    1104    1160
                        1174    1181    1340    4808#
DF30    031672           985     992     999    1006    1013    1020    1027    1034    1041    1195    1225    1232    1239
                        1247    1254    1262    1270    4812#
DF71    031676          1217    4816#
DH1     027751           822    4611#
DH11    030062           878     895     892     899     906     913     920    1074    1317    4625#
DH2     027767           829     836     843     850     857    4614#
DH20    030127           927     934    4632#
DH24    030164           955     962     969     976    1046    1053    1060    1067    1338    4637#
DH30    030221           983     990     997    4642#
DH33    030312          1004    1011    1018    1025    1032    1039    4652#
DH46    030403          1081    1088    4662#
DH50    030463          1095    1102    4671#
DH52    030520          1109    1116    1123    1130    1137    1144    1151    1275    1310    4676#
DH61    030545          1158    4620#
DH62 *  030613          1165    4687#
DH63    030640          1172    4691#
DH64    030705          1179    4698#
DH65    030751          1186    1201    1208    1324    1331    4705#
DH66    030776          1193    4709#
DH7     030035           864     871     941     948    1282    1289    1296    1303    4621#
DH71    031054          1215    4717#
DH72    031141          1223    4726#
DH73    031216          1230    1245    1252    1260    1268    4734#
DH74    031275          1237    4742#
DISPLA  001142           762#   1374#   1382#   3237#   3828#
DISPRE  000174           585#   1382
DRVQL   022115          1345    4066#
DSF1    007066          2003    2004    2006#
DSF11   007554          2110    2111    2113#
DSF13   007702          2135    2140#   2157
DSF4    007174          2028    2029    2031#
DSF6    007332          2061    2062    2064#
DSF7    007446          2086    2087    2089#
DSWR  = 177570           623#     761    1373
DT1     031352           823    4755#
DT11    031376           879     886     893     900     907     914     921    1075    1318    4759#
DT2     031356           830     837     844     851     858    4756#
DT20    031410           928     935    4761#
DT24    031420           956     963     970     977    1047    1054    1061    1068    1339    4763#
DT30    031430           984     991     998    4765#
DT33    031444          1005    1012    1019    1026    1033    1040    4767#
```

DZRPWB.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

DT46	031460	1082	1089	4769#							
DT50	031472	1096	1103	4771#							
DT52	031502	1110	1117	1124	1131	1138	1145	1152	1276	1311	4773#
DT61	031510	1159	4774#								
DT62	031520	1166	4776#								
DT63	031526	1173	4777#								
DT64	031536	1180	4779#								
DT65	031546	1187	1202	1209	1325	1332	4781#				
DT66	031554	1194	4782#								
DT7	031370	865	872	942	949	1283	1290	1297	1304	4758#	
DT71	031570	1216	4784#								
DT72	031606	1224	4787#								
DT73	031622	1231	1246	1253	1261	1269	4789#				
DT74	031636	1238	4791#								
EMTVEC=	000030	712#	1359*	1360*							
EM1	022510	821	4117#								
EM10	023040	870	4156#								
EM100	027156	1266	4539#								
EM101	027303	1274	4554#								
EM102	027352	1281	4561#								
EM103	027373	1288	4564#								
EM104	027416	1295	4568#								
EM105	027437	1302	4571#								
EM106	027462	1309	4575#								
EM107	027520	1316	4580#								
EM11	023074	877	4161#								
EM110	027575	1323	4588#								
EM111	027642	1330	4595#								
EM112	027711	1337	4602#								
EM12	023115	884	4164#								
EM13	023153	891	4170#								
EM14	023215	898	4176#								
EM15	023236	905	4179#								
EM16	023261	912	4183#								
EM17	023302	919	4186#								
EM2	022552	828	4123#								
EM20	023325	926	4190#								
EM21	023355	933	4195#								
EM22	023407	940	4200#								
EM23	023451	947	4206#								
EM24	023531	954	4215#								
EM25	023557	961	4219#								
EM26	023620	968	4225#								
EM27	023646	975	4229#								
EM3	022610	835	4128#								
EM30	023707	982	4235#								
EM31	023763	989	4243#								
EM32	024042	996	4251#								
EM33	024106	1003	4257#								
EM34	024157	1010	4264#								
EM35	024233	1017	4272#								
EM36	024277	1024	4279#								
EM37	024351	1031	4287#								
EM4	022644	842	4133#								
EM40	024426	1038	4295#								
EM41	024472	1045	4301#								

DZRPWB.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM42    024521          1052    4305#
EM43    024563          1059    4311#
EM44    024612          1066    4315#
EM45    024654          1073    4321#
EM46    024677          1080    4325#
EM47    024726          1087    4329#
EM5     022713           849    4140#
EM50    025000          1094    4336#
EM51    025052          1101    4344#
EM52    025142          1108    4354#
EM53    025207          1115    4361#
EM54    025250          1122    4367#
EM55    025317          1129    4374#
EM56    025363          1136    4381#
EM57    025427          1143    4388#
EM6     022751           856    4146#
EM60    025473          1150    4395#
EM61    025544          1157    4402#
EM62    025565          1164    4405#
EM63    025616          1171    4410#
EM64    025653          1178    4415#
EM65    025730          1185    4423#
EM66    026016          1192    4432#
EM67    026057          1199    4438#
EM7     023012           863    4152#
EM70    026167          1207    4451#
EM71    026251          1214    4460#
EM72    026351          1221    4471#
EM73    026464          1229    4484#
EM74    026554          1236    4494#
EM75    026654          1243    4505#
EM76    026744          1251    4515#
EM77    027042          1258    4526#
ERBT    005324          1722    1732#   1745    1747
ERF1    010144          2182    2183    2185#
ERF10   011346          2384    2388#   2407
ERF11   011504          2416    2417    2419#
ERF12   011630          2445    2446    2448#
ERF2    010270          2210    2211    2213#
ERF3    010416          2238    2242#   2254
ERF4    010552          2264    2269#   2288
ERF5    010720          2298    2302#   2314
ERF6    011050          2325    2329#   2348
ERF7    011216          2358    2362#   2374
ERRVEC= 000004           705#   1371    1372#   1383#   1945#   1959#   1978#   1992#   3788    3789#   3791#   3794#   3999
                        4000#   4002#   4004#
GETADR  021376          1406    3931#
GMS   = ****** U         584    1391    3176    3184    3201    3910    3911    3912    3913    3914    3916    3918    3919
                        3920    3921    3922    3923    3937    3946    3955    3964    3973    3987
GTSWR = 104406          1402    3916#
HT    = 000011           615#   3356    3397
INDEXP  021766          2828    3144    4037#   4041
IOTVEC= 000020           710#   1357#   1358#
LF    = 000012           616#   3391    3397
MASK    001202           781#
MAXCYL  001176           779#   1415#   1418#   2253    2267
```

```
MAXPAT  001200          780#  1416#  1419#  2287
PC     =x000007         636#  1406#  1450#  1506#  1564#  1566#  1572#  1601#  1658#  1659#  1727#  1735#  1763#
                       1771#  1799#  1807#  1835#  1843#  1871#  1879#  1907#  1915#  1947#  1980#  2007#  2009#
                       2015#  2032#  2034#  2049#  2065#  2068#  2074#  2090#  2092#  2098#  2114#  2116#  2122#
                       2141#  2144#  2151#  2167#  2186#  2191#  2192#  2214#  2218#  2219#  2242#  2247#  2248#
                       2269#  2274#  2275#  2302#  2307#  2308#  2329#  2334#  2335#  2362#  2367#  2368#  2388#
                       2393#  2394#  2420#  2424#  2425#  2449#  2453#  2454#  2477#  2480#  2488#  2512#  2515#
                       2524#  2542#  2550#  2564#  2578#  2587#  2590#  2606#  2615#  2629#  2637#  2651#  2660#
                       2674#  2683#  2697#  2704#  2718#  2720#  2725#  2726#  2743#  2745#  2769#  2771#  2799#
                       2832#  2860#  2861#  2871#  2913#  2914#  2979#  2980#  2999#  3013#  3023#  3037#  3065#
                       3066#  3078#  3103#  3112#  3137#  3151#  3161#  3193#  3196#  3207#  3212    3247#  3324#
                       3365#  3372#  3379#  3393#  3395#  3598#  4003#  4016#  4018#  4033#  4060#
PIRQ    = 177772        622#
PIRQVE= 000240          716#
PR0     = 000000        639#
PR1     = 000040        640#
PR2     = 000100        641#
PR3     = 000140        642#
PR4     = 000200        643#  2632
PR5     = 000240        644#  2654
PR6     = 000300        645#  2677
PR7     = 000340        646#  1422   1960   1993   2478   2487   2513   2521   2540   2549   2563   2580   2586
                       2613   2636   2658   2681
PS      = 177776        619#   620
PSW     = 177776        620#  1422#  1960#  1993#  2481#  2487#  2517#  2521#  2543#  2549#  2557#  2563#  2581#
                       2586#  2609#  2613#  2632#  2636#  2654#  2658#  2677#  2681#
PWRVEC= 000024          711#
R0PBA   006766         1971   1981#  1984   1991
R0PBA1  007024         1978   1989   1992#
R0PWC   006622         1938   1948#  1951   1958
R0PWC1  006660         1945   1956   1959#
ROCHR   = 104410       3658   3919#
ROLIN   = 104411       3732   3920#
ROOCT   = 104412       3921#  3948   3966   3989
READT   015644         3099   3100   3102#
RESREG= 104414         3323   3923#
RESRP   003152         1439   1440   1443#
RESTOR  015472         3061   3062   3064#
RESVEC= 000010          706#
RPADR   001210          787#  1411   1441   1561   1591   1723   1759   1795   1831   1867   1903   1939   1972
                       2005   2030   2063   2098   2112   2136   2184   2212   2239   2265   2299   2326   2359
                       2385   2418   2447   2474   2509   2537   2576   2604   2627   2649   2672   2695   2716
                       2740   2763   2797   2955   2988   2976   3063   3101   3135   3939   3951#  4001   4755
RPBA    = 000010        724#  1447#  1479   1520   1538#  1630   1676   1837   1842#  1850#  1856#  1975   4054
RPCA    = 000012        725#  1448#  1488   1526   1599#  1639   1682   1873   1878#  1886#  1892#  2243#  2270#
                       2987#  2990#  4055
RPCS    = 000004        722#  1412#  1445#  1461   1596#  1612   1657#  1765   1770#  1778#  1784#  2189#  2216#
                       2245#  2272#  2305#  2332#  2365#  2391#  2422#  2451#  2482#  2492   2516#  2544#  2558#
                       2582#  2631#  2653#  2676#  2699#  2719#  2995#  3074#  3105#  3140#  4015#  4052
RPDA    = 000014        726#  1449#  1497   1539   1600#  1648   1695   1909   1914#  1922#  1928#  2303#  2330#
                       2363#  2389#  2767   2991#  3139#  4056
RPDS    = 000000        720#  1413   2142#  2150#  2166#  2445#  2559#  4050
RPER    = 000002        721#  1444#  1462   1508   1595#  1603   1664   1729   1734#  1742#  1748#  4051
RPM1    = 000016        727#  1536   1688   2424   2429   2943   4057
RPM2    = 000020        728#  2140#  2159#  2164#  2165#  2483#  2545#  2608#  2744#
RPM3    = 000022        729#  1565#  1571#  1577#  1707#  2008#  2014#  2033#  2048#  2066#  2067#  2073#  2091#
```

```
                    2097#  2115#  2121#  2187#  2421#  2435   2764#  2765#  2780#  2781#  2800#  2801#  2802#
                    2803#  2804#  2805#  2806#  2807#  2808#  2809#  2810#  2811#  2812#  2813#  2814#  2815#
                    2816#  2817#  2818#  2819#  2820#  2821#  2822#  2823#  2824#  2825#  2826#  2827#  2830#
                    2831#  3146#  3147#  3148#  4017#  4023#  4038#  4039#
RPPRIO  001216       789#  1407   3975   3997#  3998
RPVEC   001212       788#  1407#  1410#  2478#  2479#  2513#  2514#  2540#  2541#  2556#  2579#  2580#  2607#
                    2630#  2652#  2675#  2698#  3957   3969#  3998#
RPWC  = 000006       723#  1446#  1470   1514   1597#  1621   1670   1801   1806#  1814#  1820#  1942   2188#
                    2215#  2271#  2304#  2364#  2390#  2450#  3104#  3138#  4053
RSTART  002774      1408#  3213
R0    =%000000       627#  1730#  1732   1743#  1746#  1766#  1768   1779#  1782#  1802#  1804   1815#  1819#
                    1838#  1840   1851#  1854#  1874#  1876   1887#  1890#  1910#  1912   1923#  1926#  1946#
                    1948   1950   1952   1957   1961#  1979#  1981   1983   1985   1990   1994#  2828#  2997#
                    3011#  3021#  3035#  3076#  3107#  3110#  3142#  3144#  3149#  3204#  3207   3270#  3271#
                    3272#  3273#  3274#  3275#  3276#  3277   3282   3287   3290#  3291   3295   3296   3315
                    3348   3349#  3350   3353#  3489   3499#  3503   3519   3520   3533#  3729   3733#  3734
                    3737   3757#  3760#  3851   3876#  3887   3888#  3889   3890#  3891#  3892#  3893#  3949#
                    3951   3967#  3969   3975#  3976#  3977#  3978#  3979#  3980   3990#  3992#  3993#  3994#
                    3995#  3996#  3997   4022   4027#  4031#  4037   4043#
R1    =%000001       628#  1731#  1744#  1767#  1780#  1803#  1816#  1839#  1852#  1875#  1888#  1911#  1924#
                    1944#  1948#  1949   1977#  1981#  1982   2868#  2869#  3287#  3299   3303   3490   3503#
                    3504   3538   3532#  3730   3735#  3743#  3745#  3747#  3750#  3753   3756#  3852   3875#
                    3999#  4002   4024
R2    =%000002       629#  1949#  1350   1953   1982#  1983   1986   3291#  3309#  3491   3502#  3506#  3509
                    3516#  3517#  3518   3523#  3531#  3731   3736#  3744#  3746#  3748#  3754   3755#  3853
                    3874#
R3    =%000003       630#  3295#  3305#  3431   3440#  3446#  3447#  3450#  3455#  3456#  3457   3466#  3492
                    3500#  3501#  3515#  3518#  3527#  3528#  3530#  3653   3655#  3656   3659#  3660   3667#
                    3668   3670   3678   3682   3684#  3690   3692   3694#  3697#  3854   3873#
R4    =%000004       631#  1411#  1412#  1413   1441#  1444#  1445#  1446#  1447#  1448#  1449#  1453   1462
                    1471   1480   1489   1498   1509   1515   1521   1527   1533   1540   1561#  1565#  1571#
                    1577#  1591#  1595#  1596#  1597#  1598#  1599#  1600#  1604   1613   1622   1631   1640
                    1649   1657#  1665   1671   1677   1683   1689   1696   1707#  1723#  1728   1734#  1742#
                    1748#  1758#  1764   1770#  1778#  1784#  1795#  1800   1806#  1814#  1820#  1831#  1836
                    1842#  1850#  1856#  1867#  1872   1878#  1886#  1892#  1903#  1908   1914#  1922#  1929#
                    1939#  1943   1972#  1976   2005#  2008#  2014#  2030#  2033#  2048#  2063#  2066#  2067#
                    2073#  2068#  2091#  2097#  2112#  2115#  2121#  2136#  2142#  2143#  2150#  2159#  2164#
                    2165#  2166#  2184#  2187#  2188#  2189#  2212#  2215#  2216#  2239#  2243#  2245#  2265#
                    2270#  2271#  2272#  2299#  2303#  2304#  2305#  2326#  2330#  2332#  2353#  2363#  2364#
                    2365#  2385#  2389#  2390#  2391#  2418#  2421#  2422#  2435   2447#  2450#  2451#  2474#
                    2482#  2483#  2492   2495#  2509#  2516#  2537#  2544#  2545#  2558#  2559#  2576#  2582#
                    2604#  2608#  2627#  2631#  2649#  2653#  2672#  2676#  2695#  2699#  2716#  2719#  2740#
                    2744#  2763#  2764#  2765#  2767   2780#  2781#  2797#  2800#  2801#  2802#  2803#  2804#
                    2805#  2806#  2807#  2808#  2809#  2810#  2811#  2812#  2813#  2814#  2815#  2816#  2817#
                    2818#  2819#  2820#  2821#  2822#  2823#  2824#  2825#  2826#  2827#  2830#  2831#  2855#
                    2867#  2876   2908#  2920#  2924   2929   2934   2943   2976#  2987#  2990#  2991#  2995#
                    3045   3063#  3074#  3101#  3104#  3105#  3135#  3138#  3139#  3140#  3146#  3147#  3148#
                    3296#  3297#  3432   3434#  3435#  3436#  3437   3438#  3452   3454#  3462#  3465#  3955
                    3872#  4015#  4017#  4023#  4038#  4039#  4050   4051   4052   4053   4054   4055   4056
                    4057   4058   4059
R5    =%000005       632#  2770#  2778#  2858#  2880   2885#  2940#  2941#  2978#  2988   2992   3003   3027
                    3047#  3289#  3293#  3312#  3319   3433   3439#  3441#  3443#  3444#  3445#  3446   3464#
                    3493   3495#  3497#  3504#  3508#  3523   3529#  3856   3871#
R6    =%000006       633#   635   1351#  1352#  1353
R7    =%000007       634#   636
SAVREG= 104413      3267   3922#
```

DZRPWB.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

```
SAVRP   022014          1450   1506   1566   1572   1601   1659   1735   1771   1807   1843   1879   1915   2009
                        2015   2034   2049   2068   2074   2092   2098   2116   2122   2144   2151   2167   2192
                        2219   2248   2275   2308   2335   2368   2394   2425   2454   2488   2524   2550   2564
                        2587   2590   2615   2637   2660   2683   2704   2720   2726   2745   2771   2832   2861
                        2871   2914   2980   2999   3013   3023   3037   3066   3078   3112   3151   4049*
SEEK    015052          2975   2979*  3048
SETOL   004130          1559   1560   1563*
SILO  = 000026           731*  2867*  2876   2920*  2934   4059
SILOT   014270          2853   2854   2856*
SILOT1  014520          2906   2907   2910*
SP    =%000006           635*  1355*  1371*  1379*  1383   1443*  1563*  1594*  1726*  1762*  1798*  1834*  1870*
                        1906*  1941*  1974*  2006*  2031*  2043*  2044*  2045   2064*  2089*  2113*  2138*  2185*
                        2197*  2198*  2199   2213*  2224*  2225*  2226   2240*  2266*  2281*  2282*  2283   2300*
                        2327*  2341*  2342*  2343   2360*  2386*  2400*  2401*  2402   2419*  2430*  2431*  2432
                        2448*  2459*  2460*  2461   2475*  2491*  2510*  2523*  2538*  2555*  2577*  2605*  2628*
                        2650*  2673*  2696*  2717*  2741*  2766*  2798*  2856*  2910*  2919*  2977*  3064*  3102*
                        3136*  3178*  3186*  3242   3256*  3259*  3299*  3303*  3348*  3349   3350*  3352   3353
                        3354*  3356   3358   3360   3366   3368*  3370*  3378*  3382   3386   3387   3391   3423*
                        3424   3425   3426*  3431*  3432*  3433*  3439   3464   3465   3466   3467*  3468*  3489*
                        3490*  3491*  3492*  3493*  3494*  3495   3498*  3511   3513*  3515   3525   3527   3529
                        3530   3531   3532   3533   3535*  3536*  3558*  3559*  3560   3567*  3570*  3571*  3575*
                        3576*  3580   3583*  3587   3589   3591   3592*  3599   3601   3603*  3604   3606*  3607*
                        3608*  3609*  3610*  3625*  3626*  3629*  3630*  3631   3635*  3636*  3637   3640   3642
                        3644*  3653*  3654*  3659   3662   3666*  3673   3677*  3696   3697   3698*  3699*  3700*
                        3727*  3728*  3729*  3730*  3731*  3733   3737*  3739   3741   3749*  3750   3752   3753*
                        3755   3756   3757   3759   3788*  3791   3793   3794   3799*  3800*  3801   3824   3825
                        3829*  3851*  3852*  3853*  3854*  3855*  3856*  3857*  3858*  3859*  3860*  3867*  3868*
                        3869*  3870*  3871   3872   3873   3874   3875   3876   3887*  3888   3898*  3899*  3939*
                        3949   3957*  3967   3980*  3990   4005   4022*  4024*  4026   4037*  4040*  4042
STACK = 001100           610*  1355   1443   1563   1594   1726   1762   1798   1834   1870   1906   1941   1974
                        2006   2031   2064   2089   2113   2138   2185   2213   2240   2266   2300   2327   2360
                        2386   2419   2448   2475   2491   2510   2523   2538   2555   2577   2605   2628   2650
                        2673   2696   2717   2741   2766   2798   2856   2910   2919   2977   3064   3102   3136
START   002364           603   1345*
START1  002372           605   1347*
START2  002400          1346   1348*
STKLMT= 177774           621*
SUCA  = 000024           730*  4058
SWR     001140           761*  1353   1373*  1375   1381*  1400   1955   1988   2938   3238   3245   3250   3254
                        3554   3591*  3783   3797   3799   3807   3814
SWREG   000176           586*  1381   1400   3554   3567
SW0   = 000001           674*
SW00  = 000001           664*   674
SW01  = 000002           663*   673
SW02  = 000004           662*   672
SW03  = 000010           661*   671
SW04  = 000020           660*   670
SW05  = 000040           659*   669
SW06  = 000100           658*   668
SW07  = 000200           657*   667   1955   1988   2938
SW08  = 000400           656*   666
SW09  = 001000           655*   665
SW1   = 000002           673*
SW10  = 002000           654*
SW11  = 004000           653*
SW12  = 010000           652*
```

```
SW13  = 020000            651#
SW14  = 040000            650#
SW15  = 100000            649#
SW2   = 000004            672#
SW3   = 000010            671#
SW4   = 000020            670#
SW5   = 000040            669#
SW6   = 000100            668#
SW7   = 000200            667#
SW8   = 000400            666#
SW9   = 001000            665#
TBITVE= 000014            707#
TIMEP   021736           2997    3011    3021    3035    3076    3107    3110    3142    3149    4022#   4031
TKVEC = 000060            714#
TPL     001206            783#   1656*   2190*   2217*   2246*   2273*   2306*   2333*   2366*   2392*   2423*   2452*   2724*
                         2996*   3075*   3106*   3109*   3141*   4014*   4023
TPVEC = 000064            715#
TRAPVE= 000034            713#   1361*   1362*
TRTVEC= 000014            708#
TSTAT   012?60           2535    2536    2538#   2539
TSTNAT  012146           2508    2512#
TSTO    003122           1425    1437#
TST1    004104           1557#
TST10   006350           1899#
TST11   006534           1935#
TST12   006700           1968#
TST13   007044           2001#
TST14   007152           2026#
TST15   007310           2059#
TST16   007424           2084#
TST17   007532           2108#
TST2    004222           1587#
TST20   007640           2132#
TST21   010122           2180#
TST22   010246           2208#
TST23   010364           2235#
TST24   010512           2261#
TST25   010666           2295#
TST26   011014           2322#
TST27   011164           2355#
TST3    005244           1719#
TST30   011312           2381#
TST31   011462           2414#
TST32   011606           2443#
TST33   011724           2470#
TST34   012112           2505#
TST35   012236           2533#
TST36   012442           2572#
TST37   012566           2600#
TST4    005430           1755#
TST40   012666           2623#
TST41   012766           2645#
TST42   013070           2668#
TST43   013172           2691#
TST44   013260           2712#
TST45   013364           2736#
```

```
TST46   013472          2759#
TST47   013666          2793#
TST5    005614          1791#
TST50   014246          2951#
TST51   014470          2904#
TST52   015022          2972#
TST53   015450          3059#
TST54   015622          3097#
TST55   015762          3131#
TST6    006000          1827#
TST7    006164          1863#
TYPOS = 104405          3304    3914#
TYPE  = 104401          1389    1395    1408    3174    3182    3199    3203    3240    3248    3279    3280    3294    3295
                        3294    3307    3311    3314    3316    3318    3321    3361    3459    3534    3565    3566    3569
                        3582    3593    3612    3665    3671    3676    3680    3685    3686    3688    3691    3695    3761
                        3763    3910#   3935    3944    3953    3962    3971    3985
TYPERR  016556          3247    3267#
TYPOC = 104402          3187    3300    3568    3911#
TYPON = 104404          3910#
TYPOS = 104403          3179    3912#   3941    3959    3982
T3P     021756          1658    2191    2218    2247    2274    2307    2334    2367    2393    2424    2453    2725    4016
                        4031#
WCBT    005674          1794    1804#   1817    1819
WRITE   016004          3133    3134    3136#
$AUTOS  001134           758#   1404#   3562    3711
$BDROR  001122           753#   1452#   1453#   1461#   1462#   1470#   1471#   1479#   1480#   1488#   1489#   1497#   1498#
                        1508#   1509#   1514#   1515#   1520#   1521#   1526#   1527#   1532#   1533#   1539#   1540#   1603#
                        1604#   1612#   1613#   1621#   1622#   1630#   1631#   1639#   1640#   1648#   1649#   1664#   1665#
                        1670#   1671#   1676#   1677#   1682#   1683#   1688#   1689#   1695#   1696#   1725#   1729#   1764#
                        1765#   1800#   1801#   1836#   1837#   1872#   1873#   1908#   1909#   1942#   1943#   1944    1975#
                        1976#   1977    4756
$BDOAT  001126           755#   1455#   1456    1457    1464#   1465    1466    1473#   1474    1475    1482#   1483    1484
                        1491#   1492    1493    1500#   1501    1502    1510#   1511    1516#   1517    1522#   1523    1528#
                        1529    1534#   1535#   1536    1541#   1542#   1543    1547#   1548    1606#   1607    1608    1615#
                        1616    1617    1624#   1625    1626    1633#   1634    1635    1642#   1643    1644    1651#   1652
                        1653    1666#   1667    1672#   1673    1678#   1679    1684#   1685    1690#   1691#   1692    1697#
                        1698#   1699    1703#   1704    1736#   1772#   1808#   1844#   1880#   1916#   1953#   1986#   2147#
                        2171#   2172#   2746#   2747    2772#   2833#   2876#   2877    2934#   2935    2931#   2982#   3000#
                        3001#   3006    3014#   3015#   3017    3024#   3025#   3030    3038#   3039#   3041    3067#   3068#
                        3069    3079#   3080#   3082    3113#   3114#   3116    3152#   3153#   3155    4756    4776    4777
                        4782
$BELL   001166           774#   3240    3266
$CHARC  017214          3363#   3373#   3380    3389#   3394#
$CKSUM  017672          3554#   3918
$CMTAG  001100           741#   1350    1351    1359    1363    1364    1365
$CM3  = 000000           771#
$CM4  = 000001           771#    772#
$CNTLC  020544          3565    3706#
$CNTLU  020537          3582    3680    3705#
$CRLF   001173           776#   1408    3248    3266    3279    3284    3294    3311    3318    3362    3397    3593    3685
                        3705    3766
$DBLK   017662          3500    3534    3542#
$DDMGN  016374          3195    3205    3211#
$DTBL   017652          3503    3538#
$ENDAD  016364           594    3207#   3261
$EMDCT  016320          1363    3197#
```

```
$ENULL  016400          3203    3214#
$EOP    016162          3162    3173#   4010
$EOPCT  016312          1363#   3194#   3198    4009#
$ERFLG  001103           744#   3235#   3266    3772    3803    3805    3811#   3832
$ERMAX  001115           750#   1366#   3805    3827#   3832
$EROR   016404          1359    3233#
$ERRPC  001116           751#   3242#   3243#   3244    3266    4755    4756    4758    4759    4761    4763    4765    4767
                        4769    4771    4773    4774    4776    4777    4779    4781    4782    4784    4787    4789    4791
$ERRTB  001244           817#   3276
$ERTTL  001112           748#   3186    3188#   3241#   3266
$ESCAPE 001164           773#   1365#   2933#   2959#   3257    3259    3266    3826#
$FILLC  001156           769#   3366    3397
$FILLS  001155           768#   3397
$GOOR   001120           752#
$UOART  001124           754#   1454#   1456#   1457    1463#   1465#   1466    1472#   1474#   1475    1481#   1483#   1484
                        1490#   1492#   1493    1499#   1501#   1502    1507#   1511    1517    1523    1529    1536    1543
                        1546#   1548    1605#   1607#   1608    1614#   1616#   1617    1623#   1625#   1626    1632#   1634#
                        1635    1641#   1643#   1644    1650#   1652#   1653    1663#   1667    1673    1679    1685    1692
                        1699    1702#   1704    1732#   1733#   1734    1739    1768#   1769#   1770    1775    1804#   1805#
                        1806    1811    1840#   1841#   1842    1847    1876#   1877#   1878    1883    1912#   1913#   1914
                        1919    1952#   1995#   2134#   2143    2145    2150    2155#   2156    2170#   2241#   2243    2252#
                        2253    2267#   2268#   2270    2296#   2287    2301#   2303    2312#   2313    2328#   2330    2346#
                        2347    2361#   2363    2372#   2373    2387#   2389    2405#   2406    2476#   2483    2495    2496#
                        2497    2511#   2742#   2744    2747    2750#   2751    2767#   2768#   2775    2782#   2783    2795#
                        2857#   2867    2877    2882#   2886#   2888#   2889#   2911#   2920    2921#   2922    2928#   2935
                        2944#   2945    2984#   3002#   3005#   3006    3016#   3017    3026#   3029#   3030    3040#   3041
                        3071#   3081#   3082    3115#   3116    3154#   3155    4756    4761    4765    4767    4769    4774
                        4777    4782    4789
$GET42  016354          3204#
$GTSWR  017742          3566#   3916
$HD     = 000000         564
$HIOCT  020732          3754#   3765#
$ICNT   001104           745#   1424#   3818#   3819    3821#   3831
$INTAG  001135           759#   3594    3711
$ITEMB  001114           749#   3244#   3266    3271
$LF     001174           777#   3266    3397    3695    3705    3766
$LPADR  001106           746#   1367#   1421#   1439#   1559#   1589#   1721#   1757#   1793#   1829#   1865#   1901#   1937#
                        1970#   2003#   2028#   2061#   2086#   2110#   2134#   2182#   2210#   2237#   2263#   2297#   2324#
                        2357#   2383#   2416#   2445#   2472#   2507#   2535#   2574#   2602#   2625#   2647#   2670#   2693#
                        2714#   2738#   2761#   2795#   2853#   2906#   2974#   3061#   3099#   3133#   3809#   3824#   3829
                        3831
$LPERR  001110           747#   1368#   1420#   1440#   1560#   1590#   1722#   1758#   1794#   1830#   1866#   1902#   1938#
                        1971#   2004#   2029#   2062#   2087#   2111#   2135#   2140#   2149#   2158#   2183#   2211#   2238#
                        2264#   2298#   2325#   2358#   2384#   2417#   2446#   2473#   2508#   2536#   2539#   2554#   2575#
                        2603#   2626#   2648#   2671#   2694#   2715#   2739#   2762#   2796#   2854#   2859#   2866#   2907#
                        2912#   2918#   2975#   3062#   3100#   3134#   3256    3809    3825#   3831
$MAIL   = ****** U      1385    1400    3250    3350    3824
$MNEM   020562          3569    3709#
$MSWR   020551          3566    3707#
$MXCNT  021212          3822    3831#
$NULL   001154           767#   3368    3397
$NWTST= 000001          1431#   1433    1553#   1555    1580#   1582    1715#   1717    1751#   1753    1787#   1789    1823#
                        1825    1859#   1861    1895#   1897    1931#   1933    1964#   1966    1997#   1999    2022#   2024
                        2055#   2057    2080#   2082    2104#   2106    2128#   2130    2176#   2178    2204#   2206    2231#
                        2233    2257#   2259    2291#   2293    2313#   2320    2351#   2353    2377#   2379    2410#   2412
                        2439#   2441    2466#   2468    2501#   2503    2529#   2531    2568#   2570    2596#   2598    2619#
```

```
                        2621    2641#   2643    2664#   2666    2687#   2689    2708#   2710    2732#   2734    2755#   2757
                        2789#   2791    2841#   2843    2894#   2896    2963#   2965    3052#   3054    3090#   3092    3124#
                        3126
$OCNT   017442          3430#   3459#   3472#
$OMODE  017444          3425#   3429#   3434    3437#   3448#   3474#
$OVER   021176          3784    3802    3810    3820    3828#
$PASS   001100           742#   1385#   3178    3191#   3192#   3214    3816    3832
$QUES   001172           775#   3266    3397    3612    3688    3705    3763    3766
$RDCHR  020154          3625#   3919
$RDDEC= ****** U        3922
$RDLIN  020274          3653#   3920
$RDOCT  020574          3727#   3921
$RDSZ = 000007          3646#
$RESRE  021252          3866#   3923
$SAVA   001230           797#   1482    1522    1633    1678    1844    1845    4034#
$SAVCA  OC.232           798#   1491    1528    1642    1684    1880    1881    4055#   4765    4784    4787
$SAVPS  001224           795#   1464    1547    1615    1703    1772    1773    2043    2197    2224    2281    2341    2400
                        2430    2459    2591    2721    2727    4052#   4759    4763    4765    4767    4769    4771    4773
                        4784    4787    4789    4791
$SAVR   001234           799#   1500    1541    1651    1697    1916    1917    2772    2773    2833    2834    2994#   4056#
                        4767    4784    4791
$SAVS   001220           793#   1567    1573    2010    2016    2035    2050    2069    2075    2093    2099    2117    2123
                        2145    2147    2152    2161    2168    2172    4050#   4758    4759    4761    4769    4771    4784
                        4787    4789    4791
$SAVER  001222           794#   1455    1510    1606    1660    1666    1736    1737    2039    2193    2220    2249    2276
                        2309    2336    2369    2395    2426    2455    4051#   4759    4763    4765    4767
$SAVMM  001236           800#   1534    1690    2962    2872    2915    2924#   2925    2929#   2930    2943#   2947    2950
                        2955    2981    3000    3014    3024    3038    3067    3079    3113    3152    4057#   4779    4781
                        4782    4784    4787    4789    4791
$SAVMC  001226           796#   1473    1516    1624    1672    1808    1809    2244#   2331#   4053#
$SETNAD 016375          3213#
$SF2A = ****** U        3924
$SAVRE  021214          3850#   3922
$SCOPE  020734          1357    3781#
$SETUP= 000127          1342#   1356    1357    1359    1361    1363    1364    1365    1367    1397    3189    3234    3253
                        3261    3549    3711    3782
$SILO   001242           802#   4059#   4779    4782
$STUP = 177777          1342#
$SUCA   001240           801#   2746    4058#   4774
$SVLEC  021150          3792    3823#
$SVPC = 000200           592#    597
$SWR = 147400            554#    564     568     569     570     571     572     573     574     772     773     774     1364
                        1365    1367    1368    1442    1562    1592    1724    1760    1796    1832    1868    1904    1940
                        1973    2006    2031    2064    2069    2113    2137    2135    2213    2240    2266    2300    2327
                        2360    2386    2419    2448    2475    2510    2538    2577    2605    2628    2650    2673    2696
                        2717    2741    2764    2798    2856    2909    2977    3064    3102    3136    3170    3190    3206
                        3212    3214    3225    3226    3227    3228    3229    3238    3245    3250    3254    3266    3773
                        3774    3775    3776    3777    3783    3795    3797    3798    3803    3804    3805    3812    3813
                        3814    3825    3829    3831
$SWRMK= 000200           554#    574     575    3777    3778    3799    3800
$TIMES  001162           772#   1364#   1442#   1592#   1724#   1760#   1796#   1832#   1868#   1904#   1940#   1973#   2137#
                        2909#   3190#   3812#   3819    3822#   3831
$TKB    001146           764#   3547    3558    3575    3629    3635
$TKS    001144           763#   3547    3556    3572    3596#   3627    3633
$TMPO   001160           771#   1737#   1738#   1739    1773#   1774#   1775    1809#   1810#   1811    1845#   1846#   1847
                        1881#   1882#   1883    1917#   1918#   1919    2773#   2774#   2775    2834#   2835#   3268#   3269#
```

```
                        4756    4758    4759    4761    4763    4765    4767    4769    4771    4773    4774    4776    4777
                        4779    4781    4782    4784    4787    4789    4791
$TN     = 000056         554#     564    1431    1442#   1553    1562#   1580    1592#   1715    1724#   1751    1760#   1787
                        1796#   1823    1832#   1859    1868#   1895    1904#   1931    1940#   1964    1973#   1997    2006#
                        2022    2031#   2055    2064#   2080    2089#   2104    2113#   2128    2137#   2176    2185#   2204
                        2213#   2231    2240#   2257    2266#   2291    2300#   2318    2327#   2351    2360#   2377    2386#
                        2410    2419#   2439    2448#   2466    2475#   2501    2510#   2529    2538#   2568    2577#   2596
                        2605#   2619    2628#   2641    2650#   2664    2673#   2687    2696#   2708    2717#   2732    2741#
                        2755    2764#   2789    2798#   2841    2856#   2894    2909#   2963    2977#   3052    3064#   3090
                        3102#   3124    3136#
$TPB    001152           766#   3386#   3397
$TPFLG  001157           770#   3344    3397
$TPS    001150           765#   3384    3397
$TRAP   021310          1361    3887#
$TRAP2  021332          3898#   3909
$TRP    = 000015        3902#   3911#   3912#   3913#   3914#   3915#   3916    3917#   3918    3919#   3920#   3921#   3922#
                        3923#   3924#
$TRPRO  021044          3892    3909#
$TSTNM  001102           743#   1423#   3189#   3237    3266    3269    3772    3801    3823#   3828    3832
$TTYIN  020530          3655    3656    3668    3686    3700    3704#
$TYPBN= ****** U        3915
$TYPDS  017446          3688#   3914
$TYPE   017000          3344#   3902    3910
$TYPEC  017150          3365    3372    3379    3384#   3385    3598
$TYPEX  017216          3390    3392    3395#
$TYPOC  017244          3428#   3911
$TYPON  017260          3427    3430#   3913
$TYPOS  017220          3423#   3912
$XTSTR  020746          3786#
$$GET4= 000000          3206#
$DFILL  017443          3424#   3428#   3438    3473#
$HOCAT= ****** U        3247    3783
.       = 031702         580#    584#    592     593#    595#    597#    601#    740#    778     1354    1367    1368    1420
                        1421    1512    1518    1524    1530    1537    1544    1668    1674    1680    1686    1693    1700
                        3177#   3214    3215#   3266    3397    3542#   3547    3704#   3705    3711#   3766    3831    3832
                        3938#   3956#
```

```
BITEST  1342#  1725  1761  1797  1833  1869  1905
COMMEN     1#   717#
ENDCOM     1#   717#
ENDPRS  1342#  3199
ERROR    611#  1459  1468  1477  1486  1495  1504  1513  1519  1525  1531  1538  1545  1550  1569
        1575  1610  1619  1628  1637  1646  1655  1662  1669  1675  1681  1687  1694  1701  1706
        1741  1777  1813  1849  1885  1921  1954  1987  2012  2018  2037  2041  2047  2052  2071
        2077  2095  2101  2119  2125  2148  2154  2163  2173  2195  2201  2222  2228  2251  2278
        2285  2311  2338  2345  2371  2397  2404  2428  2434  2457  2463  2489  2494  2525  2551
        2565  2588  2593  2616  2638  2661  2684  2705  2723  2729  2749  2777  2837  2864  2874
        2879  2917  2927  2932  2937  2949  2952  2957  2985  3008  3019  3032  3043  3072  3084
        3118  3157  4006
ESCAPE     1#   717#
GETPRI     1#   717#
GETSWR     1#   554#  717#  1396
MOVETA   733#   779
MORE.S  1342#  1437  1557  1587  1719  1755  1791  1827  1863  1899  1935  1968  2001  2026  2059
        2084  2108  2132  2180  2208  2235  2261  2295  2322  2355  2381  2414  2447  2470  2505
        2533  2572  2600  2623  2645  2668  2691  2712  2736  2759  2793  2851  29[illegible]  2972  3059
        3097  3131
MSG     1430#  1433  1579#  1582  2840#  2843  2893#  2896  2962#  2965  3051#  3054  3089#  3092  3123#
        3126
MULT       1#   717#
NEWTST     1#   717#  1431  1553  1580  1715  1751  1787  1823  1859  1895  1931  1964  1997  2022
        2055  2080  2104  2128  2176  2204  2231  2257  2291  2318  2351  2377  2410  2439  2466
        2501  2529  2568  2596  2619  2641  2664  2687  2708  2732  2755  2789  2841  2894  2963
        3052  3090  3124
POP        1#   717#  3529  3755  3871
PUSH       1#   717#  3488  3729  3851
REPORT     1#   717#
SCOPE    612#  1551  1576  1708  1749  1785  1821  1857  1893  1929  1962  1995  2019  2053  2078
        2102  2126  2174  2202  2229  2255  2289  2315  2349  2375  2408  2436  2464  2499  2526
        2566  2594  2617  2639  2662  2685  2706  2730  2753  2787  2838  2891  2960  3049  3087
        3121  3160
SETPRI     1#   717#
SETTRA  3902#  3911  3912  3913  3914  3916  3918  3919  3920  3921  3922  3923
SETUP      1#   717#  1349
SKIP       1#   717#
SLASH      1#   717#
SPACE    717#
STARS      1#   590   717#   736   778  1431  1436  1553  1556  1580  1586  1715  1718  1751  1754
        1787  1790  1823  1826  1859  1862  1895  1898  1931  1934  1964  1967  1997  2000  2022
        2025  2055  2058  2080  2083  2104  2107  2128  2131  2176  2179  2204  2207  2231  2234
        2257  2260  2291  2294  2318  2321  2351  2354  2377  2380  2410  2413  2439  2442  2466
        2469  2501  2504  2529  2532  2568  2571  2596  2599  2619  2622  2641  2644  2664  2667
        2687  2690  2708  2711  2732  2735  2755  2758  2789  2792  2841  2850  2894  2903  2963
        2971  3052  3058  3090  3096  3124  3130  3167  3221  3329  3400  3478  3546  3549  3617
        3646  3715  3769  3835  3881
SWRSLU     1#   717#  1369#
TRPTRP  3902#
TYPBIN     1#   717#
TYPDEC     1#   717#  3302
TYPEND  1342#  3174
TYPNAM     1#   717#
TYPNUM     1#   717#
TYPOCS     1#   717#  3178  3939  3957  3980
```

DZRPWB.CMB CROSS REFERENCE TABLE -- MACRO NAMES

```
TYPOCT      1#    717#  3186  3239  3567
TYPTXT      1#    717#  1389  3174  3182  3199  3934  3944  3952  3962  3970  3985
Z        1451#   1460   1469  1478  1487  1496  1602  1611  1620  1629  1638  1647
ZX       1342#   2042   2196  2223  2280  2340  2399  2429  2458
SSCMRE    734#
SSCMTM    734#    771
SSESCA      1#    717#
SSNEXT      1#    717#  1431  1553  1580  1715  1751  1787  1823  1859  1895  1931  1964  1997  2022
         2055     2080   2104  2129  2176  2204  2231  2257  2291  2318  2351  2377  2410  2439  2466
         2501     2529   2568  2596  2619  2641  2664  2687  2708  2732  2755  2789  2841  2894  2963
         3052     3090   3124
SSSET    3902#   3911   3912  3913  3914  3916  3918  3919  3920  3921  3922  3923
SSSKIP      1#    717#
.EQUAT      1#    554#   607
.HEADE      1#    554#
.KT11       1#
.SETUP      1#    554#  1342
.SUPHI      1#    554#   564
.SUPLO    575#
.$ACTI      1#    554#   588
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#    554#   578
.$CMTA      1#    554#   734
.$D820      1#
.$D820      1#
.$DIV       1#
.$EOP       1#    554#  3165
.$ERRO      1#    554#  3219
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    554#  3713
.$READ      1#    554#  3544
.$R2AZ      1#
.$SAVE      1#    554#  3833
.$S820      1#
.$S820      1#
.$SCOP      1#    554#  3767
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    554#  3879
.$TYPB      1#
.$TYPO      1#    554#  3476
.$TYPE      1#    554#  3327
.$TYPO      1#    554#  3398
.$40CA      1#
.1170       1#
```

```
ADD     1453  1462  1471  1480  1489  1498  1509  1515  1521  1527  1533  1540  1604  1613  1622
        1631  1640  1649  1665  1671  1677  1683  1689  1696  1729  1765  1801  1837  1873  1909
        1943  1976  2782  3276  3354  3426  3436  3508  3583  3592  3750
ASL     1746  1782  1818  1854  1890  1926  2155  2496  2882  3273  3274  3275  3606  3607  3608
        3743  3745  3747  3891  3976  3977  3978  3992  3993  3994  3995  3996
ASLB    3513
ASR     2888
BCC     1747  1783  1819  1855  1891  1927  3514
BCS     1388  2883  2890  3298
BEQ     1458  1467  1476  1485  1494  1503  1512  1518  1524  1530  1537  1544  1549  1574  1609
        1618  1627  1636  1645  1654  1661  1668  1674  1680  1686  1693  1700  1705  1740  1776
        1812  1848  1884  1920  1951  1984  2017  2046  2051  2076  2100  2124  2146  2153  2157
        2162  2169  2200  2227  2250  2294  2310  2344  2370  2403  2433  2462  2493  2498  2748
        2752  2776  2779  2836  2878  2926  2936  2946  2956  2983  2989  2993  3004  3007  3018
        3028  3031  3042  3070  3083  3117  3156  3205  3236  3239  3255  3258  3278  3283  3298
        3292  3306  3357  3392  3453  3563  3590  3605  3674  3738  3798  3802  3804  3808  3817
        3932  3950  3968  3991  4008
BGE     3820
BGT     3195  3460  3522  3602  3643  3740
BHI     3806
BHIS    2254  2314  2374  2784
BIC     1535  1542  1571  1691  1698  1733  1738  1769  1774  1805  1810  1841  1846  1877  1882
        1913  1918  2014  2048  2066  2097  2121  2421  2768  2774  2835  2982  3001  3015  3025
        3039  3068  3080  3114  3153  3192  3450  3559  3576  3603  3630  3636  3644  3749  3800
BIS     1456  1465  1474  1483  1492  1501  1565  1577  1607  1616  1625  1634  1643  1652  1707
        2008  2033  2067  2073  2091  2115  2187  2482  2516  2544  2558  2582  2631  2653  2676
        2699  2889  3455  3456  3516  3517  3610  4017
BIT     1413  1567  1573  1660  1955  1988  2010  2016  2035  2039  2045  2050  2069  2075  2093
        2099  2117  2123  2156  2193  2199  2220  2226  2249  2276  2283  2309  2336  2343  2369
        2395  2402  2426  2432  2435  2455  2461  2492  2497  2591  2751  2862  2872  2915  2925
        2930  2938  2947  2950  2955  3238  3245  3254  3783  3797  3807  3814
BITB    3380
BLE     3310
BLO     3669
BLOS    3657
BLT     3371  3461  3505  3521  3600  3641  3742
BMI     2728  3512
BNE     1354  1376  1387  1394  1399  1401  1414  1568  1745  1781  1817  1853  1889  1925  1956
        1989  2011  2036  2040  2070  2094  2118  2194  2221  2277  2288  2337  2348  2396  2407
        2427  2456  2592  2863  2870  2873  2881  2916  2923  2931  2939  2942  2948  2951  3046
        3246  3262  3313  3320  3351  3359  3367  3381  3388  3451  3510  3555  3561  3581  3588
        3595  3632  3638  3661  3663  3679  3683  3693  3784  3815  4025  4041
BPL     2722  3251  3345  3385  3449  3496  3526  3557  3573  3628  3634
BR      1346  1378  1390  1403  1417  1570  1958  1991  2013  2038  2072  2096  2120  2279  2339
        2398  2490  2522  2553  2589  2614  2659  2682  2703  2786  2865  2875  2884  2887  2954
        3048  3175  3183  3200  3301  3308  3322  3347  3364  3374  3383  3390  3427  3442  3463
        3507  3524  3584  3611  3613  3639  3672  3681  3687  3689  3751  3764  3786  3792  3795
        3810  3813  3936  3945  3954  3963  3972  3536
CLR     1345  1352  1364  1369  1385  1409  1410  1423  1424  1507  1663  1731  1742  1748  1767
        1778  1784  1803  1814  1820  1839  1850  1856  1875  1886  1892  1911  1922  1928  1946
        1961  1979  1994  2164  2166  2170  2171  2241  2301  2361  2481  2517  2543  2557  2581
        2609  2742  2785  2858  2959  2978  2984  3071  3188  3189  3190  3268  3270  3289  3440
        3499  3502  3570  3571  3654  3677  3735  3736  3812  3826  3933  4009
CLRB    3363  3389  3528  3684  3694  3760  3811
CMP     1353  1375  1400  1457  1466  1475  1484  1493  1502  1511  1517  1523  1529  1536  1543
        1548  1608  1617  1626  1635  1644  1653  1667  1673  1679  1685  1692  1699  1704  1739
```

DZRPWB.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	1775	1811	1847	1883	1919	1950	1983	2253	2287	2313	2347	2373	2406	2747	2775
	2783	2877	2922	2935	2945	3006	3017	3030	3041	3082	3116	3155	3261	3520	3554
	3560	3580	3587	3599	3601	3631	3637	3640	3642	3656	3668	3793	3819	4005	
CMPB	2145	2158	3356	3358	3366	3387	3391	3562	3594	3660	3678	3692	3692	3739	3741
	3801	3805													
COM	1743	1744	1779	1780	1815	1816	1851	1852	1887	1888	1923	1924	2044	2198	2225
	2282	2342	2401	2431	2460	3293	3312								
DEC	2778	2959	2941	3193	3272	3305	3309	3667	4024	4040					
DECB	3370	3373	3448	3459											
EMT	611														
HALT	584	3252	3263	3346											
INC	1386	2252	2268	2286	2312	2346	2372	2405	2750	2921	2944	3047	3191	3241	3454
	3462	3506	3609	3818											
INCB	3235	3393	3823												
IOT	612														
JMP	603	605	1425	3162	3212	4010									
JSR	1406	1450	1506	1564	1566	1572	1601	1658	1659	1727	1735	1763	1771	1799	1807
	1835	1843	1871	1879	1907	1915	1947	1980	2007	2009	2015	2032	2034	2049	2065
	2068	2074	2090	2092	2098	2114	2116	2122	2141	2144	2151	2167	2186	2191	2192
	2214	2218	2219	2242	2247	2248	2269	2274	2275	2302	2307	2308	2329	2334	2335
	2362	2367	2368	2388	2393	2394	2420	2424	2425	2449	2453	2454	2477	2480	2488
	2512	2515	2524	2542	2550	2564	2578	2587	2590	2606	2615	2629	2637	2651	2660
	2674	2683	2697	2704	2718	2720	2725	2726	2743	2745	2769	2771	2799	2828	2832
	2960	2861	2871	2913	2914	2979	2980	2997	2999	3011	3013	3021	3023	3035	3037
	3065	3066	3076	3078	3103	3107	3110	3112	3137	3142	3144	3149	3151	3161	3207
	3247	3365	3372	3379	3598	4016	4031								
MOV	1347	1351	1355	1357	1358	1359	1360	1361	1362	1363	1367	1368	1371	1372	1373
	1374	1379	1381	1382	1383	1407	1411	1412	1415	1416	1418	1419	1420	1421	1422
	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1452	1454	1455	1461
	1463	1464	1470	1472	1473	1479	1481	1482	1488	1490	1491	1497	1499	1500	1508
	1510	1514	1516	1520	1522	1526	1528	1532	1534	1539	1541	1546	1547	1559	1560
	1561	1563	1589	1590	1591	1592	1594	1595	1596	1597	1598	1599	1600	1603	1605
	1606	1612	1614	1615	1621	1623	1624	1630	1632	1633	1639	1641	1642	1648	1650
	1651	1656	1664	1666	1670	1672	1676	1678	1682	1684	1688	1690	1695	1697	1702
	1703	1721	1722	1723	1724	1726	1728	1730	1732	1734	1736	1737	1757	1758	1759
	1760	1762	1764	1766	1768	1770	1772	1773	1793	1794	1795	1796	1798	1800	1802
	1804	1806	1808	1809	1829	1830	1831	1832	1834	1836	1838	1840	1842	1844	1845
	1865	1866	1867	1868	1870	1872	1874	1876	1878	1880	1881	1901	1902	1903	1904
	1906	1908	1910	1912	1914	1916	1917	1937	1938	1939	1940	1941	1942	1944	1945
	1948	1949	1952	1953	1959	1960	1970	1971	1972	1973	1974	1975	1977	1978	1981
	1982	1985	1986	1992	1993	2003	2004	2005	2006	2028	2029	2030	2031	2043	2061
	2062	2063	2064	2086	2087	2088	2089	2110	2111	2112	2113	2134	2135	2136	2137
	2138	2139	2140	2142	2143	2149	2150	2158	2159	2165	2182	2183	2184	2185	2188
	2190	2197	2210	2211	2212	2213	2215	2217	2224	2237	2238	2239	2240	2243	2244
	2246	2263	2264	2265	2266	2267	2270	2271	2273	2281	2297	2298	2299	2300	2304
	2306	2324	2325	2326	2327	2328	2331	2332	2333	2341	2357	2358	2359	2360	2363
	2364	2366	2383	2384	2385	2386	2387	2388	2390	2392	2400	2416	2417	2418	2419
	2422	2423	2430	2445	2446	2447	2448	2450	2451	2452	2459	2472	2473	2474	2475
	2476	2478	2479	2483	2487	2491	2495	2507	2508	2509	2510	2511	2513	2514	2521
	2523	2535	2536	2537	2538	2539	2540	2541	2545	2549	2554	2555	2556	2559	2563
	2574	2575	2576	2577	2579	2580	2536	2602	2603	2604	2605	2607	2613	2625	2626
	2627	2628	2630	2632	2636	2647	2648	2649	2650	2652	2654	2658	2670	2671	2672
	2673	2675	2677	2681	2693	2694	2695	2696	2698	2714	2715	2716	2717	2724	2738
	2739	2740	2741	2746	2761	2762	2763	2764	2765	2766	2767	2770	2772	2773	2780
	2781	2795	2796	2797	2798	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809
	2810	2811	2812	2813	2814	2815	2816	2817	2818	2819	2820	2821	2822	2823	2824

DZRPWB.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	2825	2826	2827	2830	2831	2833	2834	2853	2854	2855	2856	2857	2859	2866	2867
	2868	2876	2885	2886	2906	2907	2908	2909	2910	2911	2912	2918	2919	2920	2924
	2928	2929	2933	2934	2940	2943	2974	2975	2976	2977	2981	2987	2990	2996	3000
	3002	3005	3014	3016	3024	3026	3029	3038	3040	3061	3062	3063	3064	3067	3074
	3075	3079	3081	3099	3100	3101	3102	3104	3105	3106	3109	3113	3115	3133	3134
	3135	3136	3138	3139	3140	3141	3146	3147	3148	3152	3154	3178	3186	3196	3204
	3237	3242	3256	3259	3277	3282	3287	3290	3291	3299	3303	3315	3348	3349	3353
	3368	3423	3431	3432	3433	3439	3446	3464	3465	3466	3467	3468	3489	3490	3491
	3492	3493	3494	3495	3500	3503	3523	3529	3530	3531	3532	3533	3535	3536	3567
	3591	3596	3625	3626	3653	3655	3666	3697	3698	3699	3700	3727	3728	3729	3730
	3731	3733	3734	3753	3754	3755	3756	3757	3788	3789	3791	3794	3799	3809	3821
	3822	3824	3825	3828	3829	3851	3852	3853	3854	3855	3856	3857	3858	3859	3860
	3867	3868	3869	3870	3871	3872	3873	3874	3875	3876	3887	3888	3892	3898	3899
	3939	3949	3951	3957	3967	3969	3975	3980	3990	3997	3998	3999	4000	4002	4004
	4014	4015	4022	4023	4037	4038	4039	4050	4051	4052	4053	4054	4055	4056	4057
	4056	4059													
MOVB	1366	1404	1657	2147	2172	2189	2216	2245	2272	2303	2305	2330	2365	2391	2608
	2719	2744	2991	2994	2995	3244	3269	3271	3295	3296	3350	3378	3386	3424	3425
	3428	3429	3430	3434	3437	3438	3457	3498	3501	3515	3518	3527	3558	3575	3629
	3635	3659	3664	3670	3675	3690	3737	3827	3890						
NEG	3435	3497													
NOP	1438	1558	1588	1720	1756	1792	1828	1864	1900	1936	1969	2002	2027	2060	2085
	2109	2133	2181	2209	2236	2262	2296	2323	2356	2382	2415	2444	2471	2484	2485
	2486	2506	2518	2519	2520	2534	2546	2547	2548	2560	2561	2562	2573	2583	2584
	2585	2601	2610	2611	2612	2624	2633	2634	2635	2646	2655	2656	2657	2669	2678
	2679	2680	2692	2700	2701	2702	2713	2737	2760	2794	2852	2905	2973	3060	3098
	3132	3208	3209	3210											
RESET	1348	1505	1562	1593	2160	3206									
ROL	3441	3443	3444	3445	3447	3744	3746	3748							
ROR	3297														
RTI	1380	3265	3355	3469	3537	3597	3645	3701	3758	3830	3861	3877	3900		
RTS	3324	3395	3893	4003	4018	4027	4033	4043	4060						
SUB	3243	3504													
SWAB	3979														
TRAP	3902	3911	3912	3913	3914	3916	3918	3919	3920	3921	3922	3923			
TST	1393	1398	1957	1990	2880	2988	2992	3003	3027	3045	3069	3250	3257	3319	3352
	3360	3362	3452	3509	3519	3589	3604	3662	3673	3696	3752	3759	3790	3816	3889
	3931	4001	4007	4026	4042										
TSTB	2152	2161	2721	2727	3344	3384	3511	3525	3556	3572	3627	3633	3803		
.ASCII	775	776	4066	4068	4076	4080	4089	4098	4438	4471	4526	4539			
.ASCIZ	774	777	1392	3177	3185	3202	3705	3706	3707	3709	3938	3947	3956	3965	3974
	3988	4064	4107	4117	4123	4128	4133	4140	4146	4152	4156	4161	4164	4170	4176
	4179	4183	4186	4190	4195	4200	4206	4215	4219	4225	4229	4235	4243	4251	4257
	4264	4272	4279	4287	4295	4301	4305	4311	4315	4321	4325	4329	4336	4344	4354
	4361	4367	4374	4381	4388	4395	4402	4405	4410	4415	4423	4432	4447	4451	4460
	4480	4484	4494	4505	4515	4535	4548	4554	4561	4564	4568	4571	4575	4580	4588
	4595	4602	4611	4614	4621	4625	4632	4637	4642	4652	4662	4671	4676	4680	4687
	4691	4698	4705	4709	4717	4726	4734	4742							
.BLKB	3704														
.BLKW	3542														
.BYTE	743	744	749	750	758	759	767	768	769	770	3180	3181	3214	3470	3471
	3472	3473	3702	3703	3942	3943	3960	3961	3983	3984	4797	4798	4801	4802	4805
	4806	4809	4810	4813	4814	4817	4818								
.DSABL	3614														
.ENABL	1	554	3547												
.END	4820														

.ENDC
559 571 573 574 575 587 591 595 597 611 703 717 737 741 743
771 772 773 774 775 779 803 1342 1355 1356 1359 1361 1363 1364 1365
1367 1369 1385 1392 1400 1406 1432 1433 1436 1437 1440 1442 1443 1554 1555
1556 1557 1560 1562 1581 1582 1586 1587 1590 1592 1593 1716 1717 1718 1719
1722 1724 1725 1752 1753 1754 1755 1758 1760 1761 1788 1789 1790 1791 1794
1796 1797 1824 1825 1826 1827 1830 1832 1833 1860 1861 1862 1863 1866 1868
1869 1896 1897 1898 1899 1902 1904 1905 1932 1933 1934 1935 1938 1940 1941
1965 1966 1967 1968 1971 1973 1974 1998 1999 2000 2001 2004 2006 2023 2024
2025 2026 2029 2031 2056 2057 2058 2059 2062 2063 2064 2081 2082 2083 2084 2087
2089 2105 2106 2107 2108 2111 2113 2129 2130 2131 2132 2136 2137 2138 2177
2178 2179 2180 2183 2185 2205 2206 2207 2208 2211 2213 2233 2234 2235
2238 2240 2258 2259 2260 2261 2264 2266 2300 2319
2320 2321 2322 2325 2327 2352 2353 2354 2380 2381
2384 2386 2411 2412 2413 2414 2417 2419 2448 2467
2468 2469 2470 2473 2475 2502 2503 2504 2532 2533
2536 2538 2569 2570 2571 2572 2575 2577 2605 2620
2621 2622 2623 2626 2628 2642 2643 2644 2645 2650 2666 2667 2668
2671 2673 2688 2689 2690 2691 2694 2696 2710 2711 2712 2715 2717 2733
2734 2735 2736 2739 2741 2756 2757 2758 2759 2762 2764 2790 2791 2792 2793
2796 2798 2842 2843 2850 2851 2854 2856 2895 2898 2903 2904 2907 2909 2910
2964 2965 2971 2972 2975 2977 3053 3054 3058 3059 3062 3064 3091 3092 3096
3097 3100 3102 3125 3126 3130 3131 3134 3136 3168 3170 3172 3177 3181 3182
3185 3189 3196 3198 3199 3202 3204 3206 3212 3214 3215 3222 3226 3236 3242
3247 3248 3249 3250 3261 3265 3266 3330 3350 3401 3479 3547 3548 3550 3578
3614 3618 3646 3647 3655 3657 3660 3688 3705 3711 3716 3722 3766 3770 3773
3778 3783 3795 3796 3799 3802 3803 3805 3807 3814 3818 3823 3824 3828 3831
3832 3836 3932 3938 3891 3910 3911 3912 3913 3914 3915 3916 3917 3918 3919
3920 3921 3922 3923 3924 3938 3943 3944 3947 3956 3961 3962 3965 3974 3984
3985 3988

.EQUIV
611 612 620 635 636 665 666 667 668 669 670 671 672 673 674
693 694 695 696 697 698 699 700 701 702

.EVEN
1392 3177 3185 3202 3215 3711 3938 3947 3956 3965 3974 3988 4113 4751

.IF
555 571 572 573 574 575 587 590 593 595 609 675 703 736 740
742 771 772 773 774 778 779 1342 1350 1355 1357 1359 1361 1363 1364
1365 1367 1385 1391 1397 1400 1431 1433 1436 1439 1442 1443 1553 1555 1556
1559 1562 1580 1582 1586 1589 1592 1593 1715 1717 1718 1721 1724 1725 1751
1753 1754 1757 1760 1761 1787 1789 1790 1793 1796 1797 1823 1825 1826 1829
1832 1833 1859 1861 1862 1865 1868 1869 1895 1897 1898 1901 1904 1905 1931
1933 1934 1937 1940 1941 1964 1966 1967 1970 1973 1974 1997 1998 2000 2003
2006 2022 2024 2025 2028 2031 2055 2057 2058 2061 2064 2080 2082 2083 2086
2089 2104 2106 2107 2110 2113 2128 2130 2131 2134 2137 2138 2176 2178 2179
2182 2185 2204 2206 2207 2210 2213 2231 2233 2234 2237 2240 2257 2259 2260
2263 2266 2291 2293 2294 2297 2300 2318 2320 2321 2324 2327 2351 2353 2354
2357 2360 2377 2379 2380 2383 2386 2410 2412 2413 2416 2419 2439 2441 2442
2445 2448 2466 2468 2469 2472 2475 2501 2503 2504 2507 2510 2529 2531 2532
2535 2538 2568 2570 2571 2574 2577 2596 2598 2599 2602 2605 2619 2621 2622
2625 2628 2641 2643 2644 2647 2650 2664 2666 2667 2670 2673 2687 2689 2690
2693 2696 2708 2710 2711 2714 2717 2732 2734 2735 2738 2741 2755 2757 2758
2761 2764 2789 2791 2792 2795 2798 2841 2843 2850 2853 2856 2894 2896 2903
2906 2909 2910 2963 2965 2971 2974 2977 3052 3054 3058 3061 3064 3090 3092
3096 3098 3122 3124 3126 3130 3133 3136 3167 3168 3169 3170 3171 3172 3174
3176 3180 3181 3184 3194 3197 3199 3201 3204 3206 3212 3214 3215 3221 3224
3235 3238 3245 3247 3248 3250 3254 3261 3265 3266 3329 3350 3400 3478 3546
3548 3549 3550 3578 3617 3618 3646 3654 3656 3660 3661 3704 3705 3711 3715
3768 3774 3795 3797 3798 3799 3803 3804 3805 3814 3816
3824 3825 3830 3831 3832 3835 3881 3887 3891 3902 3911 3912 3913 3914 3915

```
                3916    3918    3919    3920    3921    3922    3923    3924    3937    3942    3943    3946    3955    3960    3961
                3964    3973    3983    3984    3987
.IFF             571     573     574     575     591     595     597     609     737     740     742     771     779    1355    1431
                1432    1433    1437    1438    1442    1553    1554    1555    1557    1559    1562    1580    1581    1582    1587
                1589    1592    1715    1716    1717    1719    1722    1724    1751    1752    1753    1755    1758    1760    1787
                1788    1789    1791    1794    1796    1823    1824    1825    1827    1830    1832    1659    1860    1861    1863
                1866    1868    1895    1896    1897    1899    1902    1904    1931    1932    1933    1935    1938    1940    1964
                1965    1966    1968    1971    1973    1997    1998    1999    2001    2003    2006    2022    2023    2024    2026
                2028    2031    2055    2056    2057    2059    2061    2064    2080    2081    2082    2084    2086    2089    2104
                2105    2106    2108    2110    2113    2129    2129    2130    2132    2135    2137    2176    2177    2178    2180
                2182    2185    2204    2205    2206    2208    2210    2213    2231    2232    2233    2235    2238    2240    2257
                2258    2259    2261    2264    2266    2291    2292    2293    2295    2298    2300    2318    2319    2320    2322
                2325    2327    2351    2352    2353    2355    2358    2360    2377    2378    2379    2381    2384    2386    2410
                2411    2412    2414    2416    2419    2439    2440    2441    2443    2445    2448    2466    2467    2468    2470
                2473    2475    2501    2502    2503    2505    2508    2510    2529    2530    2531    2533    2535    2538    2568
                2569    2570    2572    2574    2577    2596    2597    2598    2600    2622    2625    2619    2620    2621    2623
                2625    2628    2641    2642    2643    2645    2647    2650    2664    2665    2666    2668    2670    2673    2687
                2688    2689    2691    2693    2696    2708    2709    2710    2712    2714    2717    2732    2733    2734    2736
                2739    2741    2755    2758    2757    2759    2762    2764    2789    2790    2791    2793    2795    2798    2941
                2942    2943    2951    2952    2956    2994    2995    2996    2904    2906    2909    2963    2964    2965    2972
                2975    2977    3052    3053    3054    3059    3061    3064    3090    3091    3092    3097    3099    3102    3124
                3125    3126    3131    3133    3136    3168    3171    3174    3181    3194    3197    3214    3222    3224    3238
                3261    3266    3330    3401    3479    3547    3550    3618    3620    3625    3646    3647    3656    3688    3704
                3716    3770    3796    3799    3815    3831    3836    3862    3888    3943    3944    3961    3962    3984    3985
.IFT            1392    3177    3185    3202    3248    3620    3625    3739    3759    3766    3813    3938    3947    3956    3965
                3974    3988
.IFTF           1392    3177    3195    3202    3247    3565    3618    3621    3735    3743    3765    3811    3938    3947    3956
                3965    3974    3988
.IIF             554     559     564     568     569     570     571     574     575     576     584     778    1356    1359    1363
                1364    1365    1367    1368    3170    3179    3187    3189    3190    3214    3215    3225    3226    3227    3228
                3229    3234    3253    3261    3266    3300    3304    3397    3547    3568    3696    3705    3711    3766    3773
                3774    3775    3776    3777    3778    3782    3812    3813    3828    3831    3832    3910    3911    3912    3913
                3914    3916    3918    3919    3920    3921    3922    3923    3940    3958    3981
.IRP             779    1342    1431    1508    1553    1580    1664    1715    1751    1787    1823    1659    1895    1931    1964
                1997    2022    2055    2080    2104    2128    2176    2204    2231    2257    2291    2318    2351    2377    2410
                2439    2466    2501    2529    2568    2596    2619    2641    2664    2687    2708    2732    2755    2789    2841
                2894    2963    3052    3080    3124    3174    3489    3529    3729    3755    3851    3871    4049
.LIST              1     554     574     584     717     771     772    1342    1369    1392    1431    1442    1553    1562    1580
                1592    1715    1724    1751    1760    1787    1796    1823    1832    1859    1868    1895    1904    1931    1940
                1964    1973    1997    2006    2022    2031    2055    2064    2080    2089    2104    2113    2128    2137    2176
                2185    2204    2213    2231    2240    2257    2266    2291    2300    2318    2327    2351    2360    2377    2386
                2410    2419    2439    2448    2466    2475    2501    2510    2529    2538    2568    2577    2596    2605    2619
                2628    2641    2650    2664    2673    2687    2696    2708    2717    2732    2741    2755    2764    2789    2798
                2841    2856    2894    2909    2963    2977    3052    3064    3090    3102    3124    3136    3177    3185    3189
                3202    3206    3261    3646    3777    3902    3910    3911    3912    3913    3914    3915    3916    3917    3918
                3919    3920    3921    3922    3923    3924    3938    3947    3956    3965    3974    3988
.MACRO             1     575     733     734    1342    1430    1451    1579    2840    2893    2962    3051    3089    3123    3902
.MCALL           554     717    1369
.NLIST             1     554     574     584     717     771     772    1342    1369    1392    1431    1442    1553    1562    1580
                1592    1715    1724    1751    1760    1787    1796    1823    1832    1859    1868    1895    1904    1931    1940
                1964    1973    1997    2006    2022    2031    2055    2064    2080    2089    2104    2113    2128    2137    2176
                2185    2204    2213    2231    2240    2257    2266    2291    2300    2318    2327    2351    2360    2377    2386
                2410    2419    2439    2448    2466    2475    2501    2510    2529    2538    2568    2577    2596    2605    2619
                2628    2641    2650    2664    2673    2687    2696    2708    2717    2732    2741    2755    2764    2789    2798
                2841    2856    2894    2909    2963    2977    3052    3064    3090    3102    3124    3136    3177    3185    3189
                3202    3206    3261    3646    3777    3902    3910    3911    3912    3913    3914    3915    3916    3917    3918
```

DZRPWB.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	3919	3920	3921	3922	3923	3924	3938	3947	3956	3965	3974	3988			
.PAGE	734	803													
.REM	1														
.REPT	584	771	2800	3976	3992										
.SBTTL	564	578	588	599	607	718	734	803	1343	1349	1397	1428	1431	1553	1580
	1715	1751	1787	1823	1859	1895	1931	1964	1997	2022	2055	2090	2104	2128	2176
	2204	2231	2257	2291	2318	2351	2377	2410	2439	2466	2501	2529	2568	2596	2619
	2641	2664	2687	2708	2732	2755	2789	2841	2894	2963	3052	3090	3124	3165	3217
	3219	3327	3398	3476	3544	3713	3767	3833	3879	3902	4115				
.TITLE	554														
.WORD	584	585	586	596	742	745	746	747	748	751	752	753	754	755	756
	757	760	761	762	771	779	780	781	782	783	787	788	789	793	794
	795	796	797	798	799	800	801	802	2829	2998	3012	3022	3036	3077	3108
	3111	3143	3145	3150	3194	3207	3213	3281	3286	3317	3394	3474	3762	3765	3909
	4032	4755	4756	4758	4759	4761	4763	4765	4767	4769	4771	4773	4774	4776	4777
	4779	4781	4782	4784	4787	4789	4791	4796	4800	4804	4808	4812	4816		

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DZRPWB.SEQ/SOL/CRF/NL:TOC/PAGNUM=DZRPWB.SML,DZRPWB.CMB
RUN-TIME: 47 64 8 SECONDS
RUN-TIME RATIO: 256/120=2.1
CORE USED:  34K  (67 PAGES)
```

N09

Spooler runtime 10 Seconds, 78 KCS, 508 disk reads, 3 disk writes, 116 pages