IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBR-E-D

PRODUCT NAME: 11/40 - 11/45 CPU PARITY TEST

DATE CREATED: MAY 1975

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: BRUCE BURGESS

CONTENTS

1. ABSTRACT

THIS PROGRAM WILL TEST PARITY ABORTS DURING CPU EXECUTION OF READ/RESTORE (DATI) AND READ/PAUSE (DATIP) MEMORY OPERATIONS. NORMAL PARITY IS GENERATED WHEN WRITING TO MEMORY (DATO) AND CHECKED FOR 'OTHER' PARITY WHEN READING FROM MEMORY (DATI OR DATIP). PARITY ABORTS ARE FORCED BY SETTING A PARITY CONTROL REGISTER FOR 'OTHER' PARITY (NOT NORMAL) BEFORE EXECUTION OF DATI OR DATIP INSTRUCTIONS.

THIS PROGRAM DOES NOT TEST MEMORY; IT TESTS THE PROCESSOR AND ASSUMES MEMORY TO BE FUNCTIONING PROPERLY. MAINDEC-11-DCMFA WILL TEST MEMORY AND SHOULD BE RUN IN CONJUNCTION WITH THIS PROGRAM TO PROVIDE A THOROUGH TEST OF PARITY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11/40 OR PDP-11/45 COMPUTER WITH CONSOLE TELETYPE, AND AN MF11 (CORE) OR MS11 (MOS) PARITY OPTION WITH ASSOCIATED PARITY MEMORY ANY WHERE WITHIN MACHINE BOUNDS

2.2 STORAGE

THIS PROGRAM REQUIRES APPROXIMATELY 3K STORAGE.

2.3 PRELIMINARY PROGRAMS

SINCE THIS PROGRAM ASSUMES MEMORY TO BE FUNCTIONING PROPERLY (AS MENTIONED IN THE ABSTRACT) IT WOULD BE WISE TO RUN MAINDEC-11-DCMFA BEFORE THIS PROGRAM.

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE PARAGRAPH 5.

4.2 STARTING ADDRESS

THE PROGRAM IS STARTED AT ADDRESS 200.

4.3 OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY USING .ABS LOADER
2. LOAD ADDRESS 200
3. SET SWITCHES, IF ANY (SEE PARAGRAPH 5.)
4. PRESS START
5. THE PROGRAM WILL LOOP AND THE TELETYPE BELL WILL RING EVERY PASS (IF SW<10>=0)

5. OPERATIONAL SWITCH SETTINGS

```
SW<15>=1...HALT ON ERROR
SW<14>=1...LOOP ON TEST
SW<13>=1...INHIBIT ERROR TYPEOUTS
SW<12>=1...ALLOW USER TO SELECT
          ...REGISTER HE DESIRES
SW<11>=1...INHIBIT ITERATIONS
            (NOT USED)
SW<10>=1...RING BELL ON ERROR
SW<10>=0...RING BELL ON PASS COMPLETE
SW<09>=1...LOOP ON ERROR
SW<08>=1...LOOP ON SPECIAL TEST SHOWN
            IN SWS<7 THRU 0>
SW<06>=1...DON'T ENABLE KT11 OPTION
            EVEN IF PRESENT
SWS<7 THRU 0>...USED IN CONJUNCTION WITH
                SW<08> DESCRIBED ABOVE
```

5.1 THE SWITCHES DEFINED ABOVE ARE SELF EXPLANATORY EXCEPT FOR THE SPECIAL COMBINATION OF SWS<06, 07 THRU 00, AND 12>. TWO (2) EXAMPLES ARE AS FOLLOWS:

(1) THE USER WISHES TO SELECT A PARTICULAR REGISTER TO UNDERGO TESTING, NOT USE THE KT11, AND LOOP ON TST37

(A) LOAD ADDRESS 200
(B) SET SWITCHES 6 AND 12
(C) HIT START
(D) THE TELETYPE WILL RESPOND BY ASKING THE USER TO 'TYPE THE REGISTER YOU DESIRE & HIT CARRIAGE RETURN' , AND WILL WAIT FOR THIS RESPONSE E.G. 172110 (NOT 772110)
(E) BEFORE TYPING A REPLY AND HITTING A CARRIAGE RETURN, PUT SW<06> AND SW<12> DOWN, SET SW<08>, AND PLACE THE VALUE 37 INTO SWS<07 THRU 00>
(F) TYPE THE RESPONSE AND HIT CARRIAGE RETURN
(G) YOU SHOULD BE LOOPING ON TST37 WHICH CAN BE EASILY VERIFIED BY EXAMINING THE CONTENTS OF $LPADR

NOTE: LOOPING ON A PARTICULAR TEST CAPABILITY WILL ONLY WORK WHEN THE USER HAS SELECTED A PARTICULAR REGISTER USING THE SW<12> OPTION

(2) THE USER WISHES TO SELECT A PARTICULAR REGISTER TO UNDERGO TESTING, USE THE KT11, AND LOOP ON TST37

USE THE SAME PROCEDURE DESCRIBED UNDER (1) ABOVE EXCEPT ONLY SET SW<12> UNDER ITEM (8)

5.2 WHEN USING THE SW<12> OPTION THE RESPONSE EXPECTED IS A 6 - DIGIT OCTAL NUMBER E.G. 172100, 172120, ETC.

IF THE USER FOR SOME REASON DOES NOT TYPE A 6-DIGIT OCTAL NUMBER E.G. 172A.....THE TELETYPE WILL CARRIAGE RETURN, LINE FEED, AND TYPE A '?' (QUESTION MARK). IT WILL SIT HERE WAITING FOR THE NUMBER TO BE TYPED CORRECTLY FOLLOWED BY A CARRIAGE RETURN.

6. SUBROUTINE ABSTRACTS

6.1 ABORT

ONCE A REGISTER IS FOUND TO BE PRESENT, THIS ROUTINE WILL SEARCH MEMORY, PERFORMING A DATI, UNTIL THE CORRESPONDING PARITY MEMORY AREA IS FOUND. THIS ROUTINE IS ONLY USED DURING THE PROGRAM TABLE CREATION.

6.2 $ACCEPT

THIS ROUTINE IN CONJUNCTION WITH $READC WILL ACCEPT AN OCTAL NUMBER FROM THE TELETYPE. THESE 2 ROUTINES ARE SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. THEY ARE USED WHEN SW<12> IS SET TO A 1 BY THE USER.

6.3 $B2OCT

THIS ROUTINE HANDLES TYPING OF BINARY TO OCTAL (ASCII) NUMBERS. IT IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. IT IS USED FOR ERROR REPORTING.

6.4 CHECKLOC

AFTER A PARITY ABORT HAS BEEN FORCED BY THE PROGRAM THIS ROUTINE WILL LOOK FOR THE CORRECT HIGH ORDER ERROR ADDRESS BITS IN THE PARITY CONTROL REGISTER AS WELL AS THE PROPER PC PUSH ON THE STACK FROM THE ABORT. ANY DISCREPANCIES ARE STORED FOR ERROR PRINTOUT.

6.5 COMPUT

THIS ROUTINE IS INITIALLY USED TO DETERMINE (TOGETHER WITH THE ABORT ROUTINE) WHERE/IF PARITY MEMORY PRESIDES FOR A SPECIFIC PARITY CONTROL REGISTER. IT CREATES A 2 LOCATION MEMORY MAP AT THE HIGH END OF A 1K BANK. FOR EXAMPLE, IF THE ADDRESS 17776 WERE ADDRESSABLE, THEN THIS ROUTINE WOULD GIVE THE FOLLOWING LOCATIONS AND CONTENTS:

LOC.	CONTENTS*	*KT11 NOT TURNED ON
17400	17402	
17402	17402	

LOC.	CONTENTS	KT11 TURNED ON
17400	23402	
17402	23402	

THESE 2 LOCATIONS AND CONTENTS WOULD THEN BE USED BY THE ABORT ROUTINE. IF A PARITY ABORT OCCURRED THEN THESE LOCATIONS AND CONTENTS WITH THEIR ASSOCIATED PARITY CONTROL REGISTER WOULD BE USED FOR SUBSEQUENT TESTING. R1 WILL ALWAYS HOLD THE FIRST LOCATION OF THE 2 LOCATION MAP.

6.6 $EOP

THIS IS THE END OF PASS ROUTINE. BEFORE THE PROGRAM LOOPS BACK TO TEST THE NEXT TABLE ENTRY (OR ITERATE ON THE CURRENT ONE) THIS ROUTINE IS ENCOUNTERED. IT IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME.

6.7 FLAGSCLR

THIS ROUTINE IS USED TO CLEAR PERTINENT FLAGS BEFORE PASSING THRU THE PROGRAM WITH ANOTHER TABLE ENTRY OR ITERATING ON THE CURRENT ENTRY.

6.8 $HLT

THIS ROUTINE CALLED (IN NUMEROUS PLACES THRU OUT THE PROGRAM) BY THE 'EMT' INSTRUCTION IS USED WHENEVER AN ERROR HAS BEEN DETECTED. THIS ROUTINE RELIES ON SWS<9,10,14,15> FOR FUNCTIONING AND IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. THE TYPERR ROUTINE WHICH TYPES OUT THE ERROR MESSAGES AND DATA HEADERS IS CALLED WITHIN THIS ROUTINE.

6.9 INITIALIZE

THIS ROUTINE WILL COMPLETELY REINITIALIZE PROGRAM FLAGS, ETC. BEFORE RESTARTING THE PROGRAM OVER AT THE BEGINNING OF THE TABLE.

6.10 PARTST

ONCE A PARITY CONTROL REGISTER HAS BEEN FOUND TO BE PRESENT THEN THIS ROUTINE IS USED TO CHECK IF THE REGISTER IN GOOD OPERATION BEFORE TESTING IS CONDUCTED.

6.11 $PWRDN

THIS ROUTINE IN CONJUNCTION WITH $PWRUP COMPRISE THE 'POWER FAIL' ROUTINES. IF THE SYSTEM GOES DOWN WHILE THE PROGRAM IS EXECUTING, GENERAL PURPOSE REGISTERS 0 THRU 5 ARE SAVED. WHEN THE SYSTEM POWERS BACK UP THE MESSAGE 'POWER' WILL BE TYPED ON THE CONSOLE TELETYPE, GENERAL PURPOSE REGISTERS 0 THRU 5 ARE RESTORED, AND THE PROGRAM WILL AUTOMATICALLY RESTART FROM THE BEGINNING. THESE 2 ROUTINES ARE SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME.

6.12 $SCOPE

THIS ROUTINE CALLED (AT THE BEGINNING AND END OF EVERY TEST) BY THE 'IOT' INSTRUCTION IS USED FOR TEST LOOPING PURPOSES. IT DEPENDS UPON SWS<8,9,11,14> FOR FUNCTIONING AND RECORDS THE STARTING ADDRESS OF EACH TEST IN '$LPADR' AS IT IS BEING ENTERED. 'LPADR' (IN THE COMMON TAG SECTION OF THE PROGRAM) MAY BE EXAMINED TO DETERMINE THE LAST TEST SUCCESSFULLY COMPLETED. THIS ROUTINE IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME.

6.13 TRAPCATCHER

A '.+2' AND 'HALT' SEQUENCE IS REPEATED FROM LOCATION 0 TO LOCATION 776 TO CATCH ANY UNEXPECTED DEVICE TRAPS. THUS, ANY UNEXPECTED TRAPS WILL HALT AT THE DEVICE TRAP VECTOR +2. WHEN/IF THIS OCCURS EXAMINATION OF THE STACK SHOULD BE THE STARTING POINT TO FIND WHERE IN THE PROGRAM YOU WERE BEFORE THE UNEXPECTED TRAP OCCURRED.

6.14 TYPERR

THIS ROUTINE CALLED WITHIN THE $HLT ROUTINE HANDLES THE ERROR MESSAGE AND DATA HEADER PRINTOUTS.

6.15 VECSET

THIS ROUTINE IS ACCESSED AT THE BEGINNING OF EVERY TEST TO SET UP THE ADDRESS OF THE SERVICE ROUTINE FOR THE PARITY ABORT VECTOR 114.

7. ERROR PRINTOUTS

```
        *** SPECIAL NOTE ***
        **********************
```

BE AWARE THAT WHEN THE PROGRAM IS BEING EXECUTED WITH MEMORY MANAGEMENT ENABLED, THE 'ACTUAL' AND 'EXPECTED' ABORT PC VALUES GREATER THAN THE LAST ADDRESS OF THE PROGRAM ARE VIRTUAL ADDRESSES. TO FIND THE PHYSICAL (OR IN REALITY) ADDRESS PULL THE OFFSET VALUE FROM THE PROGRAM TABLE DESCRIBED IN PARAGRAPH 9.2, AND DO THE ADDITION PROCEDURE OUTLINED UNDER ITEM (2), PARAGRAPH 9.3

```
        *** END OF SPECIAL NOTE ***
        **************************
```

7.1 HLT +1

```
TEST DIDN'T ABORT
PROGRAM   REGISTER     EXPECTED
   PC     UNDER TEST   ABORT PC
**      APPROPRIATE  VALUES  **
```

7.2 HLT +2

```
FATAL ERROR TO PROGRAM
PROGRAM   REGISTER
   PC     UNDER TEST
**      APPROPRIATE VALUES  **
```

NOTE: THIS ERROR REPORT WILL COME FROM 1 OF 3 TESTS IN THE 'PARTST' ROUTINE. SOMETHING WILL BE WRONG WITH BIT00 OR BIT02 OF THE PARITY CONTROL REGISTER

7.3 HLT +3

```
ABORTED INCORRECTLY
PROGRAM REGISTER          EXPECTED          ACTUAL  EXPECTED          ACTUAL
   PC  UNDER TEST         ADDR.BITS      ADDR.BITS ABORT PC       ABORT PC
**               APPROPRIATE  VALUES                                      **
```

NOTE: THIS ERROR REPORT WILL COVER A NUMBER OF OCCURRENCES:

(1) THE EXPECTED HIGH ORDER ADDRESS BITS AND THE EXPECTED ABORT PC PUSHED ON THE STACK WERE BOTH WRONG.
   A) IN THE CASE OF AN OLD MOS DESIGN WITH NO ADDRESS BITS ZEROS (0'S) WILL APPEAR UNDER THE ADDR. BITS COLUMNS.

(2) THE EXPECTED HIGH ORDER ADDRESS BITS WERE CORRECT BUT THE WRONG ABORT PC WAS PUSHED ON THE STACK.
   B) IN THIS CASE THE VALUES APPEARING UNDER THE ADDR. BITS COLUMNS WOULD BE THE SAME

(3) THE EXPECTED HIGH ORDER ADDRESS BITS WERE INCURRECT BUT THE CORRECT ABORT PC WAS PUSHED ON THE STACK
   C) IN THIS CASE THE VALUES APPEARING UNDER THE ABORT PC COLUMNS WOULD BE THE SAME

7.4 HLT +4

```
NO PARITY MEMORY FOUND BELOW 28K
REGISTER
UNDER TEST
**  APPROPRIATE VALUE  **
```

NOTE: THIS PRINTOUT WILL OCCUR FOR 1 OF 2 REASONS:

(1) WITH NO KT11 OPTION ON THE SYSTEM, A PARITY CONTROL REGISTER WAS FOUND BUT THE CORRESPONDING PARITY MEMORY WAS NOT FOUND IN LOOKING ALL THE WAY UP TO 28K, OR
(2) A KT11 OPTION IS ON THE SYSTEM WITH THE PARITY CONTROL REGISTERS' CORRESPONDING PARITY MEMORY AREA ABOVE 28K BUT THE USER DISABLED THE KT11 (DID NOT ALLOW USE) BY SETTING SW<06>.

7.5 HLT +5

```
RESET DOESN'T WORK
PROGRAM  REGISTER
   PC    UNDER TEST
**      APPROPRIATE VALUES     **
```

NOTE: IF A KT11 OPTION IS PRESENT, AND NOT DISABLED BY SETTING SW<06>, THEN THE TEST (TEST #4) INCURRING THIS PRINTOUT WILL NOT BE EXECUTED.

7.6 HLT +6

```
USER SELECTED REGISTER NOT PRESENT
PROGRAM
   PC
**  APPROPRIATE VALUE  **
```

NOTE: THIS PRINTOUT WILL COME ABOUT AS A RESULT OF USING THE SW<12> OPTION. IF IN RESPONSE TO THE MESSAGE "TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN" THE USER TYPES A NON-EXISTANT REGISTER ADDRESS THEN THE ABOVE PRINTOUT WILL OCCUR AND THE USER RESPONSE MESSAGE WILL BE REITERATED.

7.7 HLT +7

```
NO PARITY MEMORY FOUND AT ALL
REGISTER
UNDER TEST
**  APPROPRIATE VALUE  **
```

NOTE: THIS ERROR PRINTOUT COULD OCCUR FOR 1 OF 2 REASONS:

(1) THE KT11 OPTION IS PRESENT AND NOT DISABLED (USING SW<06>) INDICATING NOWHERE WAS A CORRESPONDING PARITY MEMORY AREA FOUND, OR
(2) A POSSIBLE HOLE IN MEMORY EXISTS BECAUSE WE TIMED OUT BEFORE REACHING THE SUPPOSED SYSTEM MAXIMUM CORE LOCATION

7.8 HLT +10

DIDN'T ABORT OR RECOGNIZE STACK VIOLATION
PROGRAM REGISTER EXPECTED
PC UNDER TEST ABORT PC
** APPROPRIATE VALUES **

7.9 HLT +11

ABORTED BUT STACK VIOLATION NOT RECOGNIZED
PROGRAM REGISTER
PC UNDER TEST
** APPROPRIATE VALUES **

7.10 HLT +12

STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED
PROGRAM REGISTER
PC UNDER TEST
** APPROPRIATE VALUES **

8. RESTRICTIONS

AS MENTIONED IN PARAGRAPHS 1 AND 2.3, THIS PROGRAM DOES NOT TEST MEMORY, IT TESTS THE PROCESSOR. IF PARITY MEMORY CHECKING IS WHAT YOU ARE AFTER THEN RUN MAINDEC-11-DCMFA

9. MISCELLANEOUS

9.1 EXECUTION TIME

ERROR FREE PASSES ARE ON THE ORDER OF 1 OR 2 SECONDS

9.2 PROGRAM TABLE LOCATIONS

WHEN THE SW<12> OPTION IS NOT USED THE PROGRAM WILL FIND ALL PARITY CONTROL REGISTERS AND A CORRESPONDING PARITY MEMORY LOCATION AND STORE THESE VALUES INTO A MAXIMUM 10 WORD, 4 COLUMN TABLE TO BE USED BY THE PROGRAM FOR TESTING. IF, FOR EXAMPLE, 2 PARITY CONTROL REGISTERS AND PARITY MEMORY AREAS ARE FOUND THEN PASS 1 OF THE PROGRAM WILL USE THE 1ST TABLE ENTRY INFORMATION; PASS 2 THE 2ND TABLE ENTRY INFROMATION; PASS 3 BACK TO THE 1ST TABLE ENTRY INFORMATION, ETC.

THE ABSOLUTE CORE LOCATIONS FOR TABLE ENTRYS ARE AS FOLLOWS:

$REG0(LOCATION 1340) WILL CONTAIN THE 1ST PARITY REGISTER

LOCATION 1342 UP TO 1364 WILL CONTAIN ANYMORE REGISTERS FOUND

$TMP0(LOCATION 1366) WILL CONTAIN A PARITY MEMORY LOCATION CORRESPONDING TO THE REGISTER IN $REG0

LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS

$SET0(LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED WITH THE CORRESPONDING VALUE IN $TMP0

LOCATION 1422 UP TO 1444 WILL CONTAIN THE CORRESPONDING OFFSET VALUES FOR THE OTHER REGISTERS.

NTER0(LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED WITH THE PARITY REGISTER IN $REG0

LOCATION 1452 UP TO 1474 WILL CONTAIN THE CORRESPONDING INTERLEAVE FACTORS FOR THE OTHER REGISTERS

9.3 PROGRAM TABLE SET UP WITH KT11 ENABLED

IF A KT11 OPTION IS PRESENT AND IS NOT DISABLED THRU USER SETTING OF SW<06> (SEE PARAGRAPH 5.), THE PROGRAM TABLE LOCATIONS AND CONTENTS WILL APPEAR AS DESCRIBED AND SHOWN IN THE EXAMPLE BELOW.

GIVEN: A) 172100 GOVERNING 0-8K MOS MEMORY
B) 172102 GOVERNING 8-16K CORE MEMORY
C) 172112 GOVERNING 40-48K CORE MEMORY

LOC.	REGISTER COLUMN	LOC.	MEMORY COLUMN	LOC.	OFFSET COLUMN	ILEAVE COLUMN
1340	172100	1366	23700	1420	140	2
1342	172102	1370	23700	1422	200	2
1344	172112	1372	23700	1424	2500	1
1346	0					

NOTES: (1) WHEN THE KT11 IS ENABLED THE MEMORY COLUMN CONTENTS WILL ALWAYS BE THE SAME BASE ADDRESS. (UNLESS WE HAVE MEMORY INTERLEAVING)
(2) 23700 IS A PAGE 1 ADDRESS AS SEEN BY THE KT11.

THIS VIRTUAL ADDRESS AND ITS' CORRESPONDING OFFSET VALUE WILL GIVE THE PHYSICAL ADDRESS AS FOLLOWS:

```
  VIRTUAL ADDRESS =       2  3  7  0  0
+ OFFSET VALUE    =       1  4  0
                          ----------
  PHYSICAL ADDRESS=       1  7  7  0  0
```

NOTICE THAT THE OFFSET VALUE IS TO BE SHIFTED TWICE TO THE LEFT AND THE LEFTMOST DIGIT OF THE VIRTUAL ADDRESS TO BE IGNORED BEFORE ADDING.

(3) THE PHYSICAL ADDRESS VALUE FROM ABOVE IS THE VALUE USED BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5) WHICH WILL DROP THE PHYSICAL ADDRESS DOWN SO AS NOT TO DESTROY THE .ABS LOADER (I.E. - 376 IS SUBTRACTED) THUS GIVING A PHYSICAL ADDRESS FOR THE 2 LOCATION MAP CREATION AND TESTING. THIS VALUE IS ALWAYS PRESENT IN R1. (GENERAL PURPOSE REGISTER 1)

(4) THE ZERO IN THE LAST REGISTER COLUMN LOCATION IS THE PROGRAM TABLE TERMINATION INDICATOR

(5) A '1' IN THE ILEAVE COLUMN MEANS NO INTERLEAVING
A '2' IN THE ILEAVE COLUMN MEANS 2-WAY INTERLEAVING
.
.
ETC. (UP TO 8-WAY)

9.4 PROGRAM TABLE SETUP WITH KT11 DISABLED

IF A KT11 OPTION IS PRESENT AND IS DISABLED THRU USER SETTING OF SW<06> (SEE PARAGRAPH 5.) OR NO KT11 OPTION IS PRESENT THEN, THE PROGRAM TABLE LOCATIONS AND CONTENTS WILL APPEAR AS DESCRIBED AND SHOWN IN THE EXAMPLE BELOW.

GIVEN: A) 172100 GOVERNING 0-8K MOS MEMORY
B) 172102 GOVERNING 8-16K CORE MEMORY

LOC.	REGISTER COLUMN	LOC.	MEMORY COLUMN	LOC.	OFFSET COLUMN	ILEAVE COLUMN
1340	172100	1366	17700	1420	0	1
1342	172102	1370	25700	1422	0	1
1344	0					

NOTES: (1) THE MEMORY COLUMN LOCATION CONTENTS ARE THE ACTUAL VALUES USED BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5)

(2) THE OFFSET COLUMN CONTENTS ARE NOT IN AFFECT UNLESS THE KT11 IS ENABLED (SEE PARAGRAPH 9.3)

(3) THE ZERO IN LOCATION 1344 WOULD BE THE PROGRAM TABLE TERMINATION INDICATOR.

(4) A '1' IN THE ILEAVE COLUMN MEANS NO INTERLEAVING
A '2' IN THE ILEAVE COLUMN MEANS 2-WAY INTERLEAVING
.
.
ETC. (UP TO 8-WAY)

9.5 STACK POINTER

THE STACK IS INITIALLY SET TO 1100. IT WILL REMAIN THIS VALUE FOR ALL TESTS NOT DEPENDENT ON THE STACK BEING IN PARITY MEMORY AREA. FOR EXAMPLE, A TEST CHECKING FOR A PARITY ABORT ON THE 1ST 'POP' FROM AN 'RTI' INSTRUCTION WOULD REQUIRE THE STACK TO BE IN THE PARITY MEMORY AREA CONTROLLED BY THE REGISTER UNDER TEST. IN THIS CASE THE STACK POINTER IS REPOSITIONED AND INITIALIZED TO THE 1ST ADDRESS OF THE 2 LOCATION MAP SET UP BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5).

FOR EXAMPLE, CONSIDERING THE 2ND TABLE ENTRY GIVEN IN PARAGRAPH 9.4, THE 'COMPUT' ROUTINE WOULD SET UP A 2 LOCATION MAP STARTING AT LOCATION 23302. THE STACK POINTER, FOR PERTINENT TESTS MENTIONED ABOVE, WOULD THEN BE REINITIALIZED TO 23302.

NOTE: BEWARE! IF A KT11 OPTION IS PRESENT AND ENABLED AND YOU WISH TO EXAMINE THE CONTENTS OF THE STACK (AFTER A TEST REQUIRING THE STACK TO BE REPOSITIONED ABOVE 8K HAS BEEN EXECUTED) THE STACK WOULD NOT-NOT-NOT BE AT 23302 USING THIS EXAMPLE. IT WOULD BE AT 17302 BECAUSE OF AN OFFSET VALUE. SEE THE PHYSICAL ADDRESS CALCULATION EXPLANATION UNDER PARAGRAPH 9.3.

9.6 MAINTENANCE HINT

THE FOLLOWING SHOULD BE USEFUL INFORMATION FOR 11/45 USERS WHO WISH TO EXAMINE A TEST TO ASCERTAIN STEP BY STEP WHAT THE TEST DID. THE FOLLOWING INFORMATION PRESUMES THAT THE USER HAS ACCESS TO A MAINTENANCE BOARD.

(1) MAKING SURE THAT THE PARITY REGISTER CONTROLLING THE LOWER 4K DOES NOT HAVE BIT02 SET, PROCEED TO DEPOSIT A 0 INTO THE CORE LOCATION OF THE 'SCOPE' STATEMENT AT THE BEGINNING OF THE TEST.
(2) LOAD ADDRESS 200 (SETTING SW<12> IF DESIRED) AND HIT START
(3) THE PROGRAM WILL HALT AT THE CORE LOCATION USED IN (1) ABOVE
(4) PUT THE 'SINGLE INSTRUCTION' AND 'SINGLE BUS CYCLE' SWITCHES ON THE PROCESSOR CONSOLE DOWN
(5) HIT THE CONTINUE SWITCH REPEATEDLY UNTIL THE ADDRESS OF THE INSTRUCTION THAT WAS TO CAUSE THE PARITY ABORT APPEARS IN THE ADDRESS LIGHTS.
(6) SET THE DATA DISPLAY SELECT KNOB TO DISPLAY THE CPU MICROSTATE IN BITS 7-0.
(7) LOOKING AT THE MAINTENANCE BOARD, RIGHT SIDE UP, AND TOGGLE SWITCHES ON THE RIGHT; PRESS THE BOTTOM RIGHTMOST SWITCH TO THE RIGHT.
(8) THEN JUST LIGHTLY TAP THE BOTTOM LEFTMOST SWITCH (JUST ENOUGH FOR IT TO BOUNCE BACK) REPEATEDLY. THE MICROSTATES WILL BE DISPLAYED IN BITS 7-0 OF THE CONSOLE DATA REGISTER
(9) THE MICROSTATE VALUE THAT WAS IN THE CONSOLE DATA REGISTER JUST BEFORE IT TURNED 0 WAS THE ABURT MICROSTATE.

10. PROGRAM DESCRIPTION

THE MAIN FUNCTION OF THIS PROGRAM IS TO TEST THE ABILITY OF A PARITY CONTROL REGISTER TO INTERFACE PROPERLY WITH ITS CORRESPONDING MEMORY PARITY AREA THUS ALLOWING PARITY ABORTS ON CPU EXECUTION OF DATI AND DATIP INSTRUCTIONS SET UP WITH 'NOT NORMAL' (BAD) PARITY. BASIC COMBINATIONS OF SOURCE AND DESTINATION MODES ARE TESTED TO PICK UP ALL POSSIBLE MICROSTATES AT WHICH PARITY ABORTS CAN OCCUR. ALSO TESTED ARE SUCH THINGS AS:

(A) 1ST AND 2ND 'POP' ON A MARK INSTRUCTION
(B) THE SUB INSTRUCTION
(C) A 'MOV SM0,DM0' INSTRUCTION
(D) THE 'POP' ON AN RTS INSTRUCTION
(E) 1ST AND 2ND 'POP' ON AN RTI INSTRUCTION
(F) PS AN PC FETCH INSTRUCTIONS
(G) INDEXED WORD INSTRUCTIONS (DM6,DM7,SM6)
(H) CONDITIONAL BRANCH NOT OK INSTRUCTIONS
(I) STACK VIOLATIONS IN 'RED' AND 'YELLOW' ZONES

THIS PROGRAM USED IN CONJUNCTION WITH MAINDEC-11-DCMFA SHOULD PROVIDE A PRETTY THOROUGH TEST OF PARITY.

10.1 PROGRAM FLOW DIAGRAM

```
                                .TITLE  MAINDEC-11-DCKBR-E
                                ;COPYRIGHT 1973 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                                ;PROGRAM BY BRUCE BURGESS


                                ;OPERATIONAL SWITCH SETTINGS
                                ;
                                ;       SWITCH                  USE
                                ;       ------          ----------------------
                                ;         15            HALT ON ERROR
                                ;         14            LOOP ON TEST
                                ;         13            INHIBIT ERROR TYPEOUTS
                                ;         11            INHIBIT ITERATIONS
                                ;         10            0 = BELL ON PASS COMPLETE
                                ;                       1 = BELL ON ERROR
                                ;          9            LOOP ON ERROR
                                ;          8            LOOP ON TEST IN SW<7:0>
                                        ;SPECIAL USER TYPE SWITCH SW<12>
                                        ;
                                        ;IF SET INDICATES USER INPUT
                                        ;IF CLEAR INDICATES PROGRAM FIND
                                                ;SPECIAL KT11 DISABLE SWITCH SW<00>
                                                ;
                                                ;IF SET INDICATES DON'T USE IF PRESENT
                                                ;IF CLEAR INDICATES ALLOW USE IF PRESENT

                                ;BASIC DEFINITIONS
                                ;*************
                                ;INITIAL ADDRESS OF THE STACK POINTER
001100                          STACK=  1100
                                ;*************
                                .EQUIV  EMT,HLT                         ;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE                       ;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776                          ;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177570                          SWR=    177570                          ;SWITCH REGISTER
177570                          DISPLAY=SWR

                                ;REGISTER DEFINITION
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

                                ;SWITCH DEFINITION
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

                                ;MISCELLANEOUS BIT ASSIGNMENT
                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
```



```
                                ;VECTOR ADDRESSES
                000004          EMRVEC= 4
                000010          RESVEC= 10
                000014          TBITVEC=14
                000014          TMTVEC= 14
                000014          BPTVEC= 14
                000020          IOTVEC= 20
                000024          PWRVEC= 24
                000030          EMTVEC= 30
                000034          TRAPVEC=34
                000001                  N=1                             ;DEFINE STARTING 'N' FOR, SCOPE
                                                                        ;ROUTINE WHICH PRINTS TEST NO,
                                                                        ;AND APPROPRIATE COMMENT


                000000                  .=0
                                ;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
                                ;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS

                000200                  .=200

000200  000137  001706                  JMP     @#BEGIN         ;JUMP TO STARTING ADDRESS OF PROGRAM

                000046                  .=46
000046  012042                          $ENDAD
                000052                  .=52
000052  040000                          BIT14

                000204                  .=204

                                ;KT11-D STATUS REGISTER ADDRESSES

000204  177572                  SR0:    177572
000206  177576                  SR2:    177576

                                ;KERNAL PAGE DESCRIPTOR REGISTERS

000210  172300                  KPDR0:  172300
000212  172302                  KPDR1:  172302
000214  172304                  KPDR2:  172304
000216  172306                  KPDR3:  172306
000220  172310                  KPDR4:  172310
000222  172312                  KPDR5:  172312
000224  172314                  KPDR6:  172314
000226  172316                  KPDR7:  172316

                                ;KERNAL PAGE ADDRESS REGISTERS

000230  172340                  KPAR0:  172340
000232  172342                  KPAR1:  172342
000234  172344                  KPAR2:  172344
000236  172346                  KPAR3:  172346
000240  172350                  KPAR4:  172350
000242  172352                  KPAR5:  172352
```

```
  163  000244  172354                  KWAR6:  172354
  164  000246  172356                  KWAR7:  172356
  165
  166                                  ;KT11 VECTOR ADDRESS
  167
  168  000250  000250  000252          SEGVEC: 250,252
```



```
  169
  170                                  ;*********************************************************************
  171          001100                          .=1100
  172
  173                                  ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
  174                                  ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
  175                                  ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
  176                                  ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
  177
  178                                  ;CALL:
  179                                  ;1) USING A TRAP INSTRUCTION
  180                                  ;       TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
  181                                  ;OR
  182                                  ;       TYPE
  183                                  ;       MESADR
  184
  185                                  ;2) USING A JSR INSTRUCTION
  186                                  ;       MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
  187                                  ;       JSR     PC,$TYPE                ;CALL TYPE ROUTINE
  188                                  ;       MESADDR                         ;FIRST ADRESS OF MESSAGE
  189
  190  001100  177564                  $TPS:   177564                          ;TTY PRINTER STATUS REG. ADDRESS
  191  001102  177566                  $TPB:   177566                          ;TTY PRINTER BUFFER REG. ADDRESS
  192  001104     000                  $NULL:  .BYTE   0                       ;CONTAINS NULL CHARACTER FOR FILLS
  193  001105     002                  $FILLS: .BYTE   2                       ;CONTAINS # OF FILLER CHARACTERS REQUIRED
  194  001106     000                  $TPFLG: .BYTE   0                       ;"TERMINAL AVAILABLE" FLAG (0=YES)
  195  001107     000                          .BYTE   0                       ;RESERVED
  196
  197  001110  105767  177772          $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
  198  001114  001402                          BEQ     6$                      ;BR IF YES
  199  001116  000000                          HALT                            ;HALT HERE IF NO TERMINAL
  200  001120  000407                          BR      7$                      ;LEAVE
  201  001122  010046                  6$:     MOV     R0,-(SP)                ;SAVE R0
  202  001124  017600  000002                  MOV     #2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
  203  001130  112046                  1$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
  204  001132  001005                          BNE     2$                      ;BR IF IT ISN'T THE TERMINATOR
  205  001134  005726                          TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
  206  001136  012600                          MOV     (SP)+,R0                ;RESTORE R0
  207  001140  062716  000002          7$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
  208  001144  000002                          RTI                             ;RETURN
  209  001146  004767  000026          2$:     JSR     PC,5$                   ;GO TYPE THIS CHARACTER
  210  001152  122726  000012          3$:     CMPB    #12,(SP)+               ;CHECK IF THE CHAR. TYPED WAS A  LINE FEED
  211  001156  001364                          BNE     1$                      ;GO GET NEXT CHAR. IF NOT LINE FEED
  212  001160  016746  177720                  MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
  213                                                                          ;AND THE NULL CHAR.
  214  001164  105366  000001          4$:     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
  215  001170  002770                          BLT     3$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
  216  001172  004767  000002                  JSR     PC,5$                   ;GO TYPE A NULL
  217  001176  000772                          BR      4$                      ;LOOP
  218  001200  105777  177674          5$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
  219  001204  100375                          BPL     5$
  220  001206  116677  000002  177666          MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
  221  001214  000207                          RTS     PC
  222  001216  000062                          .BLKB   62                      ;RESERVE SOME MORE CORE FOR OVERLAY CAPIBILITIES
```

```
                                ;*****************************************************
                                ;COMMON TAGS
                                
                        001300          .=1300
                                
001300  000000                $PASS:  .WORD   0                       ;CONTAINS PASS COUNT
001302  000000                $TSTNM: .WORD   0                       ;CONTAINS THE TEST NUMBER
001304  000000                $ICNT:  .WORD   0                       ;CONTAINS SUBTEST ITERATION COUNT
001306  000000                $LPADR: .WORD   0                       ;CONTAINS SCOPE LOOP ADDRESS
001310  000000                $LPERR: .WORD   0                       ;CONTAINS SCOPE RETURN FOR ERRORS
001312  000000                $ERTTL: .WORD   0                       ;CONTAINS TOTAL ERRORS DETECTED
001314      000               $ERFLG: .BYTE   0                       ;CONTAINS ERROR FLAG
001315      000                       .BYTE   0                       ;RESERVED--NOT TO BE USED
001316  000000  000000                .WORD   0,0                     ;RESERVED--NOT TO BE USED
001322      000               $ITEMB: .BYTE   0                       ;CONTAINS ITEM CONTROL BYTE
001323      000                       .BYTE   0                       ;RESERVED--NOT TO BE USED
001324  000000                $HLTAD: .WORD   0                       ;CONTAINS PC OF LAST HLT INSTRUCTION
001326  000000                $GDADR: .WORD   0                       ;CONTAINS ADDRESS OF 'GOOD' DATA
001330  000000                $BDADR: .WORD   0                       ;CONTAINS ADDRESS OF 'BAD' DATA
001332  000000                $GDDAT: .WORD   0                       ;CONTAINS 'GOOD' DATA
001334  000000                $BDDAT: .WORD   0                       ;CONTAINS 'BAD' DATA
001336  000000                $REGAD: .WORD   0                       ;CONTAINS THE ADDRESS FROM
                                                                        ;WHICH  ($REG0) WAS OBTAINED
001340  000000                $REG0:  .WORD   0                       ;CONTAINS (($REGAD)+0)
001342  000000                $REG1:  .WORD   0                       ;CONTAINS (($REGAD)+2)
001344  000000                $REG2:  .WORD   0                       ;CONTAINS (($REGAD)+4)
001346  000000                $REG3:  .WORD   0                       ;CONTAINS (($REGAD)+6)
001350  000000                $REG4:  .WORD   0                       ;CONTAINS (($REGAD)+10)
001352  000000                $REG5:  .WORD   0                       ;CONTAINS (($REGAD)+12)
001354  000000                $REG6:  .WORD   0                       ;CONTAINS (($REGAD)+14)
001356  000000                $REG7:  .WORD   0                       ;CONTAINS (($REGAD)+16)
001360  000000                $REG10: .WORD   0                       ;CONTAINS (($REGAD)+20)
001362  000000                $REG11: .WORD   0                       ;CONTAINS (($REGAD)+22)
001364  000000                $REG12: .WORD   0                       ;CONTAINS (($REGAD)+24)
001366  000000                $TMP0:  .WORD   0                       ;USER DEFINED
001370  000000                $TMP1:  .WORD   0                       ;USER DEFINED
001372  000000                $TMP2:  .WORD   0                       ;USER DEFINED
001374  000000                $TMP3:  .WORD   0                       ;USER DEFINED
001376  000000                $TMP4:  .WORD   0                       ;USER DEFINED
001400  000000                $TMP5:  .WORD   0                       ;USER DEFINED
001402  000000                $TMP6:  .WORD   0                       ;USER DEFINED
001404  000000                $TMP7:  .WORD   0                       ;USER DEFINED
001406  000000                $TMP10: .WORD   0                       ;USER DEFINED
001410  000000                $TMP11: .WORD   0                       ;USER DEFINED
001412  000000                $TMP12: .WORD   0                       ;USER DEFINED
                                ;THE FOLLOWING TAG(S) ARE USER DEFINED
001414  000000                $TMPAD: .WORD   0
001416  000000                $SETAD: .WORD   0
001420  000000                $SET0:  .WORD   0                       ;THESE LOCATIONS CONTAIN THE
001422  000000                $SET1:  .WORD   0                       ;APPROPRIATE OFFSET VALUES
001424  000000                $SET2:  .WORD   0                       ;FOR THE PARITY CONTROL
001426  000000                $SET3:  .WORD   0                       ;REGISTERS WHEN MEMORY
001430  000000                $SET4:  .WORD   0                       ;MANAGEMENT IS ENABLED
```



```
001432  000000                $SET5:  .WORD   0                       ;DURING PROGRAM EXECUTION
001434  000000                $SET6:  .WORD   0
001436  000000                $SET7:  .WORD   0
001440  000000                $SET10: .WORD   0
001442  000000                $SET11: .WORD   0
001444  000000                $SET12: .WORD   0
001446  000000                NIERAD: .WORD   0
001450  000000                NIER0:  .WORD   0                       ;THESE LOCATIONS CONTAIN THE
001452  000000                NIER1:  .WORD   0                       ;APPROPRIATE INTERLEAVE FACTORS
001454  000000                NIER2:  .WORD   0                       ;FOR THE PARITY CONTROL
001456  000000                NIER3:  .WORD   0                       ;REGISTERS (IF ANY)
001460  000000                NIER4:  .WORD   0
001462  000000                NIER5:  .WORD   0
001464  000000                NIER6:  .WORD   0
001466  000000                NIER7:  .WORD   0
001470  000000                NIER10: .WORD   0
001472  000000                NIER11: .WORD   0
001474  000000                NIER12: .WORD   0
001476  000000                NEWSTK: .WORD   0
                                ;END OF USER DEFINED TAG(S)
```

```
                                ;THE FOLLOWING TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;THIS INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($HLTAD).
                                ;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

001500                          $ERRTB:
                                ;NOTE:  ALL NUMBERS ARE TYPED AS 6-DIGIT OCTAL NUMBERS

                                ;ITEM 1
001500  013752                          EM1                     ;TEST DIDN'T ABORT
001502  015005                          DH5                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
                                                                ;EXPECTED ABORT PC
001504  015142                          DT5                     ;$HLTAD,PARITY,$GDDAT
001506  000000                          0
                                ;ITEM 2
001510  013776                          EM2                     ;FATAL ERROR TO PROGRAM
001512  014470                          DH1                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
001514  015106                          DT1                     ;$HLTAD, PARITY
001516  000000                          0
                                ;ITEM 3
001520  014027                          EM3                     ;ABORTED INCORRECTLY
001522  014604                          DH4                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
                                                                ;EXPECTED BITS 5 THRU 11
                                                                ;ACTUAL BITS 5 THRU 11
                                                                ;EXPECTED ABORT PC
                                                                ;ACTUAL ABORT PC
001524  015124                          DT4                     ;$HLTAD,PARITY,$GDADR,$BDADR,$GDDAT,$BDDAT
001526  000000                          0
                                ;ITEM 4
001530  014055                          EM4                     ;NO PARITY MEMORY FOUND BELOW 28K
001532  014540                          DH2                     ;REGISTER UNDER TEST
001534  015114                          DT2                     ;PARITY
001536  000000                          0
                                ;ITEM 5
001540  014120                          EM5                     ;RESET DOESN'T WORK
001542  014470                          DH1                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
001544  015106                          DT1                     ;$HLTAD, PARITY
001546  000000                          0
                                ;ITEM 6
001550  014145                          EM6                     ;USER SELECTED REGISTER NOT PRESENT
001552  014566                          DH3                     ;PROGRAM PC
001554  015120                          DT3                     ;$HLTAD
001556  000000                          0
```



```
                                ;ITEM 7
001560  014212                          EM7                     ;NO PARITY MEMORY FOUND AT ALL
001562  014540                          DH2                     ;REGISTER UNDER TEST
001564  015114                          DT2                     ;PARITY
001566  000000                          0
                                ;ITEM 10
001570  014252                          EM10                    ;DIDN'T ABORT OR RECOGNIZE
                                                                ;STACK VIOLATION
001572  015005                          DH5                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
                                                                ;EXPECTED ABORT PC
001574  015142                          DT5                     ;$HLTAD, PARITY, $GDDAT
001576  000000                          0
                                ;ITEM 11
001600  014326                          EM11                    ;ABORTED BUT STACK VIOLATION
                                                                ;NOT RECOGNIZED
001602  014470                          DH1                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
001604  015106                          DT1                     ;$HLTAD, PARITY
001606  000000                          0
                                ;ITEM 12
001610  014403                          EM12                    ;STACK VIOLATION PICKED UP BUT
                                                                ;ABORT NOT RECOGNIZED
001612  014470                          DH1                     ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
001614  015106                          DT1                     ;$HLTAD, PARITY
001616  000000                          0
```

```
                              ;*****************************************************************
                              ;
                              ;SYSTEM PARITY REGISTER NOTES FOR MF11 AND MS11
                              ;
                              ;*****************************************************************

                              ;BIT ASSIGNMENTS FOR THE MF11 PARITY REGISTER IS AS FOLLOWS:
                              ;
                              ;BIT15          PARITY ERROR
                              ;BITS 11-5      ERROR ADDRESS           ;HIGH ORDER ADDRESS BITS
                                                                      ;OF ADDRESS OF MOST RECENT ERROR
                                                                      ;(BITS 17 THRU 11)
                              ;BIT02          WRITE                   ;NORMAL PARITY (ODD) WHEN CLEAR
                                                                      ;OTHER PARITY (EVEN) WHEN SET
                              ;BIT00          ERROR ACTION ENABLE     ;NO ACTION WHEN CLEAR
                              ;                                       ;TRAP TO VECTOR 114 WHEN SET
                              ;
                              ;NOTE: THE ABOVE BITS ARE READ/WRITE AND CAN BE CLEARED BY 'INIT' (EXCEPT BITS 11-5)
                              ;
                              ;\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
                              ;
                              ;BIT ASSIGNMENTS FOR THE MS11 PARITY REGISTER IS AS FOLLOWS:
                              ;
                              ;BIT15          PARITY ERROR
                              ;BIT02          WRITE                   ;NORMAL PARITY (EVEN) WHEN CLEAR
                                                                      ;OTHER PARITY (ODD) WHEN SET
                              ;BIT00          ERROR ACTION ENABLE     ;NO ACTION WHEN CLEAR
                                                                      ;TRAP TO VECTOR 114 WHEN SET
                              ;
                              ;NOTE:  THERE ARE NO ERROR ADDRESS BITS IN THE CURRENT MS11 PARITY REGISTER
                              ;       HOWEVER, THERE WILL BE IN A LATER VERSION WHICH WILL BE
                              ;       HANDLED PROPERLY BY THIS PROGRAM

                              ;**********************************************************************
                              ;**********************************************************************
                              ;
                              ;SPECIAL NOTE----THERE ARE 2 GENERAL PURPOSE REGISTERS USED IN THE
                              ;                 PROGRAM FOR SPECIFIC CIRCUMSTANCES.  THEY ARE:
                              ;
                              ;               R1 - WILL ALWAYS CONTAIN THE 1ST ADDRESS OF THE 2
                              ;                       LOCATION MAP USED FOR TESTING.
                              ;                       THE CONTENTS OF R1 IS DETERMINED BY THE 'COMPUT'
                              ;                       ROUTINE SHOWN FURTHER DOWN.
                              ;                       EXAMINATION OF R1 WILL TELL YOU WHERE IN PARITY
                              ;                       MEMORY TESTING IS BEING CONDUCTED.
                              ;
                              ;                R5 - WILL ALWAYS CONTAIN THE ADDRESS OF THE ROUTINE
                              ;                       FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS.
                              ;
                              ;**********************************************************************
                              ;**********************************************************************
```



```
                              ;*****************************************************************
                              ;
                              ;MISCELLANEOUS COMMON PARITY VARIABLES AND FLAGS
                              ;
                              ;*****************************************************************
 001620  000114               INTVEC: 114                         ;PARITY INTERRUPT VECTOR ADDRESS
 001622  000000               PARITY: 0                           ;CONTAINS PARITY REGISTER IN USE
 001624  000000               PSPCORZONES:    0                   ;FLAG TO INDICATE TO 'CHECKLOC'
                                                                  ;ROUTINE THAT A PS OR PC FETCH
                                                                  ;OR A ZONE ABORT WAS DONE
                                                                  ;0 = NO;  1 = YES
 001626  000000               MSREGFLAG:      0                   ;INDICATES PARITY TYPE
                                                                  ;0 = CORE ; 1 = MOS
 001630  000000               USERTYPE:       0                   ;INDICATES USER SELECTION OF
                                                                  ;PARITY REGISTER
                                                                  ;0 = PROGRAM FIND
                                                                  ;1 = USER SELECTION
 001632  000000               BLKCNT: 0                           ;CONTAINS THE NUMBER OF CONSEC-
                                                                  ;UTIVE LOCATIONS TO BE TESTED
                                                                  ;DURING PROGRAM TABULATION TO
                                                                  ;COVER CASES OF MEMORY INTER-
                                                                  ;LEAVING
 001634  000000               RESTOREBASE:    0                   ;HOLDS PAGE 1 ADDRESS
                                                                  ;OR CURRENT MEMORY ADDRESS FOR
                                                                  ;RESTORATION DURING
                                                                  ;RUNNING OF PROGRAM.  IT IS USED IF
                                                                  ;WE HAVE CHECKED CONSECUTIVE
                                                                  ;LOCATIONS WITHOUT AN ABORT BEFORE
                                                                  ;GOING TO NEXT OFFSET WHICH WILL
                                                                  ;PUT US IN ANOTHER BANK
 001636  000000               LEAFCNT:        0                   ;CONTAINS THE NO. OF ABORTS
                                                                  ;ENCOUNTERED IN DETERMINATION OF
                                                                  ;AN INTERLEAVE FACTOR
 001640  000000               MEMAD:  0                           ;CONTAINS A BASE ADDRESS OR A
                                                                  ;CURRENT MEMORY ADDRESS USED IN
 001642  000000               CPU40:  0                           ;FLAG TO INDICATE PROCESSOR
                                                                  ;0 = 11/45; 1 = 11/40
                                                                  ;PARITY TABLE CREATION

                              ;THE FOLLOWING TABLE IS USED TO DETERMINE THE
                              ;INTERLEAVE FACTOR FOR THE CONTROL REGISTERS
 001644                       INTERTABLE:
 001644  000003  000001       3,1                     ;3 ABORTS ON 3 CONSECUTIVE LOCS. = 1 WAY LEAVE
 001650  000006  000002       6,2                     ;3 ABORTS ON 6 CONSECUTIVE LOCS. = 2 WAY LEAVE
 001654  000011  000003       9.,3                    ;3 ABORTS ON 9 CONSECUTIVE LOCS. = 3 WAY LEAVE
 001660  000014  000004       12.,4                   ;3 ABORTS ON 12 CONSECUTIVE LOCS. = 4 WAY LEAVE
 001664  000017  000005       15.,5                   ;3 ABORTS ON 15 CONSECUTIVE LOCS. = 5 WAY LEAVE
 001670  000022  000006       18.,6                   ;3 ABORTS ON 18 CONSECUTIVE LOCS. = 6 WAY LEAVE
 001674  000025  000007       21.,7                   ;3 ABORTS ON 21 CONSECUTIVE LOCS. = 7 WAY LEAVE
 001700  000030  000010       24.,8.                  ;3 ABORTS ON 24 CONSECUTIVE LOCS. = 8 WAY LEAVE
 001704  000000               0                       ;END OF TABLE TERMINATOR
```

```
                                BEGIN:
001706  012706  001100                  MOV     #STACK,SP               ;SETUP THE STACK POINTER
001712  012737  012062  000020          MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
001720  012737  000340  000022          MOV     #340,@#IOTVEC+2         ;LEVEL 7
001726  005067  177350                  CLR     $TSTNM                  ;INITIALIZE THE TEST NUMBER
001732  012737  012634  000030          MOV     #$HLT,@#EMTVEC          ;EMT VECTOR FOR HLT(ERROR) ROUTINE
001740  012737  000340  000032          MOV     #340,@#EMTVEC+2         ;LEVEL 7
001746  012737  013556  000034          MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
001754  012737  000340  000036          MOV     #340,@#TRAPVEC+2        ;LEVEL 7
001762  012737  013606  000024          MOV     #$PWRDN,@#PWRVEC        ;POWER FAILURE VECTOR
001770  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;LEVEL 7
001776  005067  177276                  CLR     $PASS                   ;CLEAR THE PASS COUNT
002002  005067  177276                  CLR     $ICNT                   ;INITIALIZE THE ITERATION COUNTER
002006  005067  010276                  CLR     $TIMES                  ;INITIALIZE NUMBER OF ITERATIONS
002012  105067  177276                  CLRB    $ERFLG                  ;CLEAR THE ERROR FLAG
002016  005067  177270                  CLR     $ERTTL                  ;CLEAR THE ERROR COUNT
002022  005067  010746                  CLR     $ESCAPE                 ;CLEAR THE ESCAPE ON ERROR ADDRESS

002026  005037  001630                  CLR     @#USERTYPE              ;SET USER SELECTION INDICATOR
                                                                        ;TO ZERO INDICATING PROGRAM
                                                                        ;TABULATION
002032  005037  002304                  CLR     @#$KT11                 ;CLEAR KT11 PRESENCE FLAG
002036  005037  001420                  CLR     @#$SET0                 ;CLEAR THE OFFSET
002042  005037  001422                  CLR     @#$SET1                 ;TABLE LOCATIONS FOR
002046  005037  001424                  CLR     @#$SET2                 ;THE KT11 OPTION
002052  005037  001426                  CLR     @#$SET3
002056  005037  001430                  CLR     @#$SET4
002062  005037  001432                  CLR     @#$SET5
002066  005037  001434                  CLR     @#$SET6
002072  005037  001436                  CLR     @#$SET7
002076  005037  001440                  CLR     @#$SET10
002102  005037  001442                  CLR     @#$SET11
002106  005037  001444                  CLR     @#$SET12
002112  005037  001450                  CLR     @#NTER0                 ;CLEAR THE INTERLEAVE TABLE
002116  005037  001452                  CLR     @#NTER1                 ;ENTRY LOCATIONS
002122  005037  001454                  CLR     @#NTER2
002126  005037  001456                  CLR     @#NTER3
002132  005037  001460                  CLR     @#NTER4
002136  005037  001462                  CLR     @#NTER5
002142  005037  001464                  CLR     @#NTER6
002146  005037  001466                  CLR     @#NTER7
002152  005037  001470                  CLR     @#NTER10
002156  005037  001472                  CLR     @#NTER11
002162  005037  001474                  CLR     @#NTER12
002166  005037  001636                  CLR     @#LEAFCNT               ;CLEAR NO. OF ABORTS PER NO. OF
                                                                        ;CONSECUTIVE LOCS. TESTED FLAG
002172  005037  001642                  CLR     @#CPU40                 ;CLEAR PROCESSOR INDICATOR FLAG
002176  013746  000004                  MOV     @#4,-(SP)               ;SAVE CONTENTS OF LOC. 4
002202  013746  000010                  MOV     @#10,-(SP)              ;SAVE CONTENTS OF LOC. 10
002206  012737  002226  000010          MOV     #1$,@#RESVEC            ;SET UP FOR 'SPL' TRAP ADDRESS
002214  012737  000340  000012          MOV     #340,@#RESVEC+2         ;SET UP FOR 'SPL' TRAP PS
002222  000237                          SPL     7                       ;ATTEMPT TO SET A PRIORITY LEVEL
002224  000403                          BR      2$                      ;BRANCH INDICATING WE ARE ON AN
```



```
                                                                        ;11/45 PROCESSOR
002226  022626                  1$:     CMP     (SP)+,(SP)+             ;RESET THE STACK FROM TRAP
002230  005237  001642                  INC     @#CPU40                 ;SET FLAG INDICATING WE ARE ON
                                                                        ;AN 11/40 PROCESSOR
002234  012637  000010          2$:     MOV     (SP)+,@#10              ;RESTORE CONTENTS OF LOC. 10
002240  005037  000012                  CLR     @#12                    ;RESTORE TRAPCATCHER LOC. 12
002244  012737  002324  000004          MOV     #KTTIMEOUT,@#ERRVEC     ;SET UP KT TIMEOUT ADDRESS
002252  012737  000340  000006          MOV     #340,@#ERRVEC+2         ;SET UP KT TIMEOUT PS
002260  005777  175720                  TST     @SR0                    ;KT11 ARE YOU THERE?
002264  005077  175714                  CLR     @SR0                    ;YES - INITIALIZE IT IN CASE
                                                                        ;USER DOESN'T WANT IT
002270  013700  177570                  MOV     @#SWR,R0                ;GET SWR CONTENTS
002274  006300                          ASL     R0                      ;MOVE BIT06 TO BIT07 POSITION
002276  105700                          TSTB    R0                      ;KT11 PRESENT (OBVIOUSLY) IF
                                                                        ;WE REACH THIS INSTRUCTION
                                                                        ;DOES USER WANT IT?
002300  100412                          BMI     GO                      ;BRANCH IF NO
002302  005327                          DEC     (PC)+                   ;YES - SET KT11 FLAG
002304  000000                  $KT11:  0                               ;CONTAINS A -1 IF KT11 OPTION
                                                                        ;IS PRESENT
002306  004737  013124                  JSR     PC,@#$SIZE              ;SEE HOW MUCH MEMORY IS AVAILABLE
002312  005077  175712                  CLR     @#KPAR0                 ;CLEAR PAGE 0 OFFSET REGISTER
002316  005277  175662                  INC     @SR0                    ;TURN ON MEMORY MANAGEMENT
002322  000401                          BR      GO                      ;SKIP NEXT INSTRUCTION
002324  022626                  KTTIMEOUT:      CMP     (SP)+,(SP)+     ;RESET THE STACK FROM TIMEOUT
                                                                        ;KT11 NOT PRESENT, THEREFORE
                                                                        ;ONLY DO BELOW 28K
002326  012637  000004          GO:     MOV     (SP)+,@#4               ;RESTORE CONTENTS OF LOC. 4
002332  005037  000006                  CLR     @#6                     ;RESTORE CONTENTS OF LOC. 6
002336  004337  011506                  JSR     R3,@#INITIALIZE         ;SET UP TO BEGIN TESTING
002342  016703  177046                  MOV     $TMPAD,R3               ;SET UP FOR MEMORY TABLE CREATION
002346  016702  176764                  MOV     $REGAD,R2               ;SET UP FOR PARITY TABLE CREATION
002352  016700  177040                  MOV     $SETAD,R0               ;SET UP FOR OFFSET TABLE CREATION
                                                                        ;THIS TABLE ONLY HAS EFFECT IF
                                                                        ;MEMORY MGMT IS TURNED ON
002356  016705  177064                  MOV     NTERAD,R5               ;SET UP FOR INTERLEAVE TABLE
                                                                        ;CREATION (8 - WAY INTERLEAVE
                                                                        ;CAPABILITY EXISTS)
                                ;*****************************************************************
                                ;
                                ;LET'S DETERMINE IF SEVERAL REGISTERS EXIST, FOR EXAMPLE,
                                ;
                                ;       172100 GOVERNING CORE MEMORY    0 - 8K
                                ;   &   172102 GOVERNING MOS MEMORY     8 - 16K
                                ;   &   172112 GOVERNING CORE MEMORY    40 - 56K
                                ;
                                ;IF WE WANT TO PRESELECT ONE OF THEM OR CREATE A TABLE OF ALL THOSE
                                ;AVAILABLE AND CARRY ON TESTING FROM THE TABLE
                                ;
                                ;NOTE:  SEE DOCUMENT CONCERNING TABLE APPEARANCES AS A
                                ;       FUNCTION OF MEMORY MANAGEMENT (KT11 OPTION) BEING
                                ;       ENABLED OR DISABLED DURING PROGRAM EXECUTION
                                ;
                                ;*****************************************************************
```

```
  593  002362  012737  010000  177570          BIT     #BIT12,@#SWR            ;DOES THE USER WISH TO SELECT THE
  594                                                                           ;REGISTER?
  595  002370  001445                          BEQ     FINDONE                 ;BRANCH IF NO
  596  002372                          MSGTYP:
  597  002372  104400  002400                  TYPE    ,.+4                    ;TYPE ASCIZ STRING
  598  002376  000433                          BR      6$                      ;GET OVER THE ASCII
  599                                           ;.ASCIZ      <15><12>"TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN  "
  600  002466                          6$:
  601  002466  104406  001340                  ACCEPT,$REG0                    ;PICK UP THE DESIRED REGISTER
  602                                                                           ;FROM THE TELETYPE AND STORE
  603                                                                           ;IN FIRST TABLE LOCATION
  604  002472  005237  001630                  INC     @#USERTYPE              ;SET FLAG INDICATING USER SELECTION
  605  002476  013746  000004                  MOV     @#4,-(SP)               ;SAVE CONTENTS OF LOC. 4
  606  002502  000404                          BR      NEXT1                   ;SKIP THE NEXT INSTRUCTIONS
  607  002504  012712  172100          FINDONE:        MOV     #172100,(R2)    ;MOVE 1ST POSSIBLE PARITY
  608                                                                           ;REGISTER INTO $REG0
  609  002510  013746  000004                  MOV     @#4,-(SP)               ;PUSH CONTENTS OF LOC.4 ONTO STACK
  610  002514  012704  001644          NEXT1:  MOV     #INTERTABLE,R4          ;INITIALIZE INTERLEAVE TABLE
  611                                                                           ;POINTER
  612  002520  005737  002304                  TST     @#SKT11                 ;KT11 ARE YOU THERE?
  613  002524  001402                          BEQ     5$                      ;BRANCH IF NO
  614  002526  005077  175500                  CLR     @#KPAR1                 ;RESET PAGE 1 ADDRESS REGISTER
  615                                                                           ;BEFORE TESTING NEXT PARITY
  616                                                                           ;CONTROL REGISTER
  617  002532  012737  003144  000004  5$:     MOV     #NOREG,@#4              ;SET PARITY TIMEOUT VECTOR SERVICE ADDRESS
  618  002540  022712  172136                  CMP     #172136,(R2)            ;IS THE ADDRESS IN BOUNDS?
  619  002544  100002                          BPL     12$                     ;BRANCH IF YES
  620  002546  000137  003172                  JMP     @#NOMORE                ;OTHERWISE - TERMINATE TABLE!
  621  002552  005772  000000          12$:    TST     @(R2)                   ;YES - IS THIS REGISTER PRESENT?
  622  002556  004737  003222                  JSR     R7,@#PARTST             ;CHECK OUT FOR FATAL ERRORS
  623                                           ;
  624                                           ;
  625                                           ;WE HAVE CHECKED OUT THE REGISTER AND FOUND IT TO BE WORKING PROPERLY
  626                                           ;NOW WE WILL FIND ITS ASSOCIATED PARITY MEMORY, IF IT EXISTS!!
  627                                           ;
  628                                           ;
  629  002562  012737  003112  000004          MOV     #PAKCORE,@#4            ;SET MEMORY TIMEOUT VECTOR
  630                                                                           ;SERVICE ADDRESS
  631  002570  012737  013700  001640          MOV     #13700,@#MEMAD          ;SET UP A STARTING ADDRESS
  632  002576  011437  001632                  MOV     (R4),@#BLKCNT           ;SET A COUNTER FOR CONSECUTIVE
  633                                                                           ;LOCATION CHECKS TO COVER MEMORY
  634                                                                           ;INTERLEAVING
  635  002602  005737  002304                  TST     @#SKT11                 ;SHOULD I LOOK ABOVE 28K?
  636  002606  100013                          BPL     1$                      ;BRANCH IF NO
  637  002610  062737  010000  001640          ADD     #10000,@#MEMAD          ;STEP UP TO A PAGE 1 BASE ADDRESS
  638                                                                           ;IF MEMORY MANAGEMENT TURNED ON
  639  002616  013737  001640  001634          MOV     @#MEMAD,@#RESTOREBASE   ;SAVE PAGE 1 BASE ADDRESS
  640  002624  062710  000140          2$:     ADD     #140,(R0)               ;SET UP AN OFFSET FOR KPAR1
  641  002630  011077  175376                  MOV     (R0),@#KPAR1            ;SET OFFSET IN PAGE 1 REGISTER
  642  002634  000406                          BR      9$                      ;SKIP NEXT 2 INSTRUCTIONS
  643  002636  062737  004000  001640  1$:     ADD     #4000,@#MEMAD           ;STEP UP TO NEXT BANK
  644  002644  013737  001640  001634          MOV     @#MEMAD,@#RESTOREBASE   ;SAVE INITIAL MEMORY ADDRESS
  645  002652  005777  176762          9$:     TST     @#MEMAD                 ;IS THIS MEMORY AVAILABLE?
  646  002656  013713  001640                  MOV     @#MEMAD,(R3)            ;YES - STORE THIS MEMORY LOCATION
```



```
  647  002662  004737  003430                  JSR     R7,@#ABORT              ;NOW LET'S SEE IF IT'S PARITY
  648                                                                           ;MEMORY CORRESPONDING TO THE
  649                                                                           ;PARITY REGISTER WE'VE FOUND
  650  002666  005737  002304                  TST     @#SKT11                 ;KT11 ARE YOU THERE?
  651  002672  100034                          BPL     4$                      ;BRANCH IF NO
  652  002674  005337  001632                  DEC     @#BLKCNT                ;DECREASE CONSECUTIVE
  653                                                                           ;LOCATION COUNTER
  654  002700  005737  001632                  TST     @#BLKCNT                ;ARE WE DONE CHECKING
  655                                                                           ;CONSECUTIVE LOCATIONS?
  656  002704  001404                          BEQ     6$                      ;BRANCH IF YES
  657  002706  062737  000002  001640          ADD     #2,@#MEMAD              ;STEP UP 1 LOCATION
  658  002714  000756                          BR      9$                      ;GO BACK TO TEST WITH THIS
  659                                                                           ;LOCATION
  660  002716  013737  001634  001640  6$:     MOV     @#RESTOREBASE,@#MEMAD   ;RESTORE PAGE 1 BASE ADDRESS
  661                                                                           ;BEFORE GOING BACK TO INCREASE
  662                                                                           ;OFFSET
  663  002724  062704  000004                  ADD     #4,R4                   ;STEP TABLE POINTER UP FOR
  664                                                                           ;NEXT VALUE OF CONSECUTIVE
  665                                                                           ;LOCATION TO BE CHECKED
  666  002730  005714                          TST     (R4)                    ;ARE THERE ANY MORE?
  667  002732  001405                          BEQ     10$                     ;BRANCH IF NO
  668  002734  011437  001632                  MOV     (R4),@#BLKCNT           ;STORE THIS VALUE OF CONSECUTIVE
  669                                                                           ;LOCATION CHECKS
  670  002740  005037  001636                  CLR     @#LEAFCNT               ;CLEAR INTERLEAVE VALUE HOLDER
  671                                                                           ;BEFORE RETESTING
  672  002744  000742                          BR      9$                      ;GO BACK TO TEST WITH THIS VALUE
  673                                                                           ;OF CONSECUTIVE LOCATIONS
  674  002746  012704  001644          10$:    MOV     #INTERTABLE,R4          ;INITIALIZE INTERLEAVE TABLE
  675                                                                           ;POINTER
  676  002752  011437  001632                  MOV     (R4),@#BLKCNT           ;RESET THE CONSECUTIVE
  677                                                                           ;LOCATION COUNTER
  678  002756  005037  001636                  CLR     @#LEAFCNT               ;CLEAR INTERLEAVE VALUE HOLDER
  679                                                                           ;BEFORE RETESTING
  680  002762  000720                          BR      2$                      ;GO BACK TO INCREASE OFFSET
  681                                                                           ;AND TEST
  682  002764  022737  157700  001640  4$:     CMP     #157700,@#MEMAD         ;ARE WE UP TO 28K YET?
  683  002772  001434                          BEQ     8$                      ;BRANCH IF YES
  684  002774  005337  001632                  DEC     @#BLKCNT                ;DECREASE CONSECUTIVE
  685                                                                           ;LOCATION COUNTER
  686  003000  005737  001632                  TST     @#BLKCNT                ;ARE WE DONE CHECKING CONSEC-
  687                                                                           ;UTIVE LOCATIONS?
  688  003004  001404                          BEQ     7$                      ;BRANCH IF YES
  689  003006  062737  000002  001640          ADD     #2,@#MEMAD              ;STEP UP 1 LOCATION
  690  003014  000716                          BR      9$                      ;GO BACK TO TEST WITH THIS
  691                                                                           ;LOCATION
  692  003016  013737  001634  001640  7$:     MOV     @#RESTOREBASE,@#MEMAD   ;RESTORE INITIAL MEMORY
  693                                                                           ;ADDRESS BEFORE GOING BACK TO
  694                                                                           ;STEP UP TO NEXT BANK
  695  003024  062704  000004                  ADD     #4,R4                   ;STEP TABLE POINTER UP FOR
  696                                                                           ;NEXT VALUE OF CONSECUTIVE
  697                                                                           ;LOCATIONS TO BE CHECKED
  698  003030  005714                          TST     (R4)                    ;ARE THERE ANY MORE?
  699  003032  001405                          BEQ     11$                     ;BRANCH IF NO
  700  003034  011437  001632                  MOV     (R4),@#BLKCNT           ;STORE THIS VALUE OF
```

```
                                                        ;CONSECUTIVE LOCATION CHECKS
003040  005037  001636          CLR     @#LEAFCNT       ;CLEAR INTERLEAVE VALUE HOLDER
                                                        ;BEFORE RETESTING
003044  000702                  BR      9$              ;GO BACK TO STEP UP TO THE
                                                        ;NEXT BANK TO CONDUCT TESTING
003046  012704  001644  11$:    MOV     #INTERTABLE,R4  ;INITIALIZE INTERLEAVE
                                                        ;TABLE POINTER
003052  011437  001632          MOV     (R4),@#BLKCNT   ;RESET THE CONSECUTIVE
                                                        ;LOCATION COUNTER
003056  005037  001636          CLR     @#LEAFCNT       ;CLEAR INTERLEAVE VALUE HOLDER
                                                        ;BEFORE RETESTING
003062  000665                  BR      1$              ;GO BACK TO STEP UP TO NEXT
                                                        ;BANK TO CONDUCT TESTING
003064  011237  001622  8$:     MOV     (R2),@#PARITY   ;STORE THE BAD REGISTER WITH
                                                        ;NO PARITY MEMORY
003070  104004                  HLT     +4              ;NO PARITY MEMORY FOUND
                                                        ;BELOW 28K!!!!!!!
003072  005737  001630          TST     @#USERTYPE      ;DID USER SELECT REGISTER?
003076  001402                  BEQ     3$              ;BRANCH IF NO
003100  000137  002372          JMP     @#MSGTYP        ;GO BACK TO RETYPE MESSAGE FOR
                                                        ;USER RESPONSE
003104  062712  000002  3$:     ADD     #2,(R2)         ;PLACE NEXT POSSIBLE REGISTER
                                                        ;INTO SAME TABLE LOCATION
003110  000601                  BR      NEXT1           ;GO BACK TO TEST THIS REGISTER
003112  062626          PARCORE: CMP     (SP)+,(SP)+     ;RESET STACK FROM MEMORY TIMEOUT
003114  011237  001622          MOV     (R2),@#PARITY   ;STORE THE REGISTER THAT EN-
                                                        ;COUNTERED A POSSIBLE HOLE IN
                                                        ;MEMORY
003120  104007                  HLT     +7              ;A POSSIBLE HOLE IN MEMORY EXISTS
                                                        ;WITH NO PARITY BELOW IT!!!!!!!
003122  005737  001630          TST     @#USERTYPE      ;DID USER SELECT REGISTER?
003126  001402                  BEQ     4$              ;BRANCH IF NO
003130  000137  002372          JMP     @#MSGTYP        ;GO BACK TO RETYPE MESSAGE FOR
                                                        ;USER RESPONSE
003134  062712  000002  4$:     ADD     #2,(R2)         ;PLACE NEXT POSSIBLE REGISTER
                                                        ;INTO SAME TABLE LOCATION
003140  000137  002514          JMP     @#NEXT1         ;GO BACK TO TEST THIS REGISTER
003144  062626          NOREG:  CMP     (SP)+,(SP)+     ;RESET STACK FROM REGISTER TIMEOUT
003146  005737  001630          TST     @#USERTYPE      ;DID THE USER SELECT THE REGISTER?
003152  001403                  BEQ     1$              ;BRANCH IF NO
003154  104006                  HLT     +6              ;YES - USER SELECTED REGISTER NOT
                                                        ;PRESENT ON SYSTEM
003156  000137  002372          JMP     @#MSGTYP        ;GO BACK TO RETYPE MESSAGE
003162  062712  000002  1$:     ADD     #2,(R2)         ;STEP UP TO NEXT PARITY REGISTER
                                                        ;AT SAME TABLE LOCATION
003166  000137  002514          JMP     @#NEXT1         ;PREVIOUS PARITY REGISTER NOT
                                                        ;PRESENT - SEE IF THE NEXT ONE IS
003172  012637  000004  NOMORE: MOV     (SP)+,@#4       ;RESTORE CONTENTS OF LOC. 4
003176  005012                  CLR     (R2)            ;ALL DONE TABLE CREATION
                                                        ;END IT WITH A '0'
003200  000137  003620          JMP     @#START         ;START RUNNING PROGRAM WITH
                                                        ;TABLE CONTENTS
                                ;*********************************************
                                ;
```



```
                                ;THE FOLLOWING ROUTINE WILL CREATE A 2 LOCATION MEMORY MAP AT
                                ;THE HIGH END OF A 1K CORE SECTION.  THIS 2 LOCATION MAP WILL
                                ;INITIALLY BE USED TO DETERMINE WHERE/IF PARITY MEMORY
                                ;RESIDES AND LATER FOR SUBSEQUENT PROGRAM TESTING OF A REGISTER
                                ;
                                ;*********************************************
003204  162700  000376  COMPUT: SUB     #376,R0         ;DROP DOWN SO AS NOT TO
                                                        ;DESTROY .ABS LOADER
003210  010001                  MOV     R0,R1           ;R1 CONTAINS BEGINNING ADDRESS
                                                        ;OF MEMORY MAP
003212  020020                  CMP     R0,(R0)+        ;STEP R0 TO NEXT ADDRESS
003214  010011                  MOV     R0,@R1          ;1ST MEMORY LOCATION
003216  011110                  MOV     @R1,@R0         ;2ND MEMORY LOCATION
003220  000203                  RTS     R3              ;RETURN TO TEST A DATI
                                                        ;WITH CONTENTS OF THESE 2 LOCS.

                                ;*********************************************
                                ;
                                ;THIS ROUTINE WILL CHECK IF THE PARITY REGISTER IS STATICALLY IN
                                ;GOOD OPERATION FOR TESTING TO BE CONDUCTED
                                ;
                                ;*********************************************
003222  011267  176374  PARTST: MOV     (R2),PARITY     ;GET PARITY REGISTER TO BE USED
                                ;*********************************************
                                ;TEST 1 SET BIT0 (USED) OF PARITY REGISTER
                                ;*********************************************
003226  000004          TST1:   SCOPE
003230  052777  000001  176364          BIS     #BIT0,@PARITY
003236  052777  000001  176356          BIT     #BIT0,@PARITY   ;DID IT SET?
003244  001001                  BNE     .+4             ;YES
003246  104002                  HLT     +2              ;NO - FATAL ERROR TO PROGRAM!
                                ;*********************************************
                                ;TEST 2 CLEAR BIT0 (USED) OF PARITY REGISTER
                                ;*********************************************
003250  000004          TST2:   SCOPE
003252  042777  000001  176342          BIC     #BIT0,@PARITY
003260  052777  000001  176334          BIT     #BIT0,@PARITY   ;DID IT CLEAR?
003266  001401                  BEQ     .+4             ;YES
003270  104002                  HLT     +2              ;NO - FATAL ERROR TO PROGRAM!
                                ;*********************************************
                                ;TEST 3 SET AND CLEAR BIT2 (USED) OF PARITY REGISTER
                                ;*********************************************
003272  000004          TST3:   SCOPE
003274  052777  000004  176320          BIS     #BIT2,@PARITY
003302  052777  000004  176312          BIT     #BIT2,@PARITY   ;DID IT SET?
003310  001001                  BNE     .+4             ;YES
003312  104002                  HLT     +2              ;NO - FATAL ERROR TO PROGRAM!
003314  042777  000004  176300          BIC     #BIT2,@PARITY
003322  052777  000004  176272          BIT     #BIT2,@PARITY   ;DID IT CLEAR?
003330  001401                  BEQ     .+4             ;YES
003332  104002                  HLT     +2              ;NO - FATAL ERROR TO PROGRAM!
                                ;*********************************************
                                ;TEST 4 TEST RESET ON BITS 0, 2 AND 15
                                ;*********************************************
```

```
                                TST4:   SCOPE
003336  005737  002304                  TST     @#KT11                  ;KT11 ON?
003342  100415                          BMI     WHICH1                  ;BRANCH IF YES AND DON'T DO
                                                                        ;THIS TEST BECAUSE THE 'RESET'
                                                                        ;WILL CLOBBER SEGMENTATION
003344  052777  100005  176250          BIS     #100005,@PARITY
003352  000005                          RESET                           ;EXPECT BITS 0, 2 AND 15 TO CLEAR
003354  032777  100005  176240          BIT     #100005,@PARITY         ;DID THEY CLEAR?
003362  001404                          BEQ     .+12                    ;YES
003364  042777  100005  176230          BIC     #100005,@PARITY         ;NO - CLEAR OUT REGISTER AS A
                                                                        ;PRECAUTION
003372  104005                          HLT     +5                      ;RESET DOESN'T WORK
                                ;*******************************************************
                                ;TEST 5 WHICH OPTION IS ABOUT TO BE TESTED
                                ;*******************************************************
003374  000004                  TST5:   SCOPE

003376  052777  007740  176216  WHICH1: BIS     #7740,@PARITY           ;IS AN OLD MS11 OPTION
                                                                        ;WITH NO ADDRESS BITS
                                                                        ;ABOUT TO BE TESTED?
003404  032777  007740  176210          BIT     #7740,@PARITY           ;ADDRESS BITS ABLE TO BE SET?
003412  001402                          BEQ     1$                      ;BRANCH IF NO INDICATING MS11
003414  000004                          SCOPE
003416  000207                          RTS     R7                      ;RETURN TO NORMAL FLOW
003420  005237  001626          1$:     INC     @#MSREGFLAG             ;SET FLAG INDICATING MS11 OPTION
                                                                        ;WITH NO ADDRESS BITS
003424  000004                          SCOPE
003426  000207                          RTS     R7                      ;RETURN TO NORMAL FLOW

                                ;*******************************************************
                                ;
                                ;THE FOLLOWING ROUTINE WILL TAKE EACH 1K BANK OF MEMORY
                                ;THAT IS AVAILABLE AND PERFORM A DATI IN IT
                                ;TO DETERMINE IF PARITY EXISTS THERE.  THIS ROUTINE IS
                                ;ONLY USED DURING TABLE CREATION
                                ;
                                ;*******************************************************
003430  010546                  ABORT:  MOV     R5,-(SP)                ;SAVE R5 CONTENTS ON STACK
003432  010046                          MOV     R0,-(SP)                ;SAVE R0 CONTENTS ON STACK
003434  011300                          MOV     (R3),R0                 ;GET THE MEMORY LOCATION
                                                                        ;JUST DETERMINED
003436  004337  003204                  JSR     R3,@#COMPUT             ;COMPUTE AN AREA IN THIS BANK
                                                                        ;FOR DETERMINING PARITY MEMORY
003442  011267  176154                  MOV     (R2),PARITY             ;GET THE PARITY REGISTER JUST
                                                                        ;FOUND AND TEST WITH IT
                                ;*******************************************************
                                ;               TEST A DATI IN THIS BANK
                                ;*******************************************************
                                ;                 11/45 **** ROM STATE 221 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
003446  012705  011450                  MOV     #VECSET,R5              ;SET UP SERVICE ROUTINE ADDRESS
003452  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
003454  003500                          ONETRY                          ;ROUTINE ADDRESS
```



```
003456  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
003460  010010                          MOV     R0,@R0                  ;DO THE DATO
003462  010030                          MOV     R0,@(R0)+               ;DO A DATI
003464  042777  000005  176130          BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
003472  012600                          MOV     (SP)+,R0                ;RESTORE R0 CONTENTS
003474  012605                          MOV     (SP)+,R5                ;RESTORE R5 CONTENTS
003476  000207                          RTS     R7                      ;NOT PARITY MEMORY!
                                                                        ;RETURN TO TEST AT NEXT
                                                                        ;INCREMENT
003500  042777  000005  176114  ONETRY: BIC     #BIT2!BIT0,@PARITY      ;WE HAVE PARITY MEMORY - PROCEED
003506  016600  000004                  MOV     4(SP),R0                ;RESTORE R0 CONTENTS
003512  016605  000006                  MOV     6(SP),R5                ;RESTORE R5 CONTENTS
003516  005237  001636                  INC     @#LEAFCNT               ;INCREMENT INTERLEAVE COUNTER
003522  042737  000003  001636          CMP     #3,@#LEAFCNT            ;3 ABORTS REACHED?
003530  001403                          BEQ     1$                      ;BRANCH IF YES
003532  062706  000010                  ADD     #10,SP                  ;BYPASS JUNK ON STACK
003536  000207                          RTS     R7                      ;RETURN TO TEST AT NEXT INCREMENT
003540  042626                  1$:     CMP     (SP)+,(SP)+             ;POP STACK BACK FROM PARITY ABORT
003542  012600                          MOV     (SP)+,R0                ;RESTORE R0 CONTENTS
003544  012605                          MOV     (SP)+,R5                ;RESTORE R5 CONTENTS
003546  005726                          TST     (SP)+                   ;POP STACK ONCE FOR ABORT ROUTINE
                                                                        ;ENTRY
003550  005737  001630                  TST     @#USERTYPE              ;DID USER TYPE IN REGISTER?
003554  001404                          BEQ     2$                      ;BRANCH IF NO
003556  016415  000002                  MOV     2(R4),(R5)              ;SET INTERLEAVE VALUE INTO
                                                                        ;TABLE
003562  000137  003620                  JMP     @#START                 ;AND LOCK ON THE USER SELECTED
                                                                        ;REGISTER FOR TESTING
003566  005723                  2$:     TST     (R3)+                   ;USER DIDN'T SELECT - SO STEP UP
                                                                        ;TO NEXT MEMORY TABLE LOCATION
003570  005720                          TST     (R0)+                   ;STEP UP TO NEXT OFFSET TABLE
                                                                        ;LOCATION - THIS TABLE WILL ONLY
                                                                        ;BE APPLICABLE IF MEMORY MGMT
                                                                        ;IS TURNED ON
003572  012212                          MOV     (R2)+,(R2)              ;SET NEXT POSSIBLE REGISTER INTO
003574  062712  000002                  ADD     #2,(R2)                 ;NEXT REGISTER TABLE LOCATION
003600  016425  000002                  MOV     2(R4),(R5)+             ;SET INTERLEAVE VALUE INTO
                                                                        ;TABLE
003604  005037  001636                  CLR     @#LEAFCNT               ;RESET NO. OF ABORTS COUNTER
003610  005037  001626                  CLR     @#MSREGFLAG             ;CLEAR PARITY TYPE INDICATOR
003614  000137  002514                  JMP     @#NEXT1                 ;GO BACK TO CHECK NEXT POSSIBLE
                                                                        ;PARITY REGISTER


                                ;*******************************************************
                                ;*******************************************************
                                ;*******************************************************
                                ;*******************************************************
                                ;
                                ;
                                ;IF WE HAVE REACHED THIS POINT IN THE PROGRAM THEN,
                                ;
                                ;@REG0(LOCATION 1340) WILL CONTAIN THE FIRST PARITY REGISTER
```

```
                                ;
                                ;               LOCATION 1342 UP TO 1364 WILL CONTAIN ANY MORE
                                ;               REGISTERS FOUND
                                ;$TMP0(LOCATION 1366) WILL CONTAIN AN ADDRESS IN THE BANK
                                ;                     THAT HAS PARITY ASSOCIATED WITH THE PARITY
                                ;                     REGISTER IN $REG0
                                ;
                                ;               LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING
                                ;               MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS
                                ;$SET0(LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED
                                ;                     WITH THE CORRESPONDING VALUE IN $TMP0
                                ;
                                ;               LOCATION 1422 TO 1444 WILL CONTAIN THE CORRESPONDING
                                ;               OFFSET VALUES FOR THE OTHER REGISTERS
                                ;
                                ;$NTER0(LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED
                                ;                      WITH THE PARITY REGISTER IN $REG0
                                ;
                                ;               LOCATION 1452 TO 1474 WILL CONTAIN THE CORRESPONDING
                                ;                INTERLEAVE FACTORS FOR THE OTHER REGISTERS
                                ;
                                ;NOTE:  TO UNDERSTAND THE TABLE SETUP SEE THE DOCUMENT
                                ;
                                ;
                                ;*************************************************************
                                ;*************************************************************
                                ;*************************************************************
                                ;*************************************************************


                                ;*************************************************************
                                ;
                                ;THE REST OF THE PROGRAM WILL TEST ALL PARITY ABORTS
                                ;USING EITHER THE PARITY REGISTER TYPED BY THE USER OR THOSE
                                ;ENCOUNTERED IN THE TABLE PREVIOUSLY GENERATED.
                                ;
                                ;*************************************************************
003620  005777  175512          START:  TST     @$REGAD                 ;ARE WE AT END OF TABLE?
003624  001401                          BEQ     1$                      ;BRANCH IF YES
003626  000402                          BR      2$                      ;OTHERWISE SKIP NEXT INSTRUCTION
003630  000137  011464          1$:     JMP     @#RESTART               ;SET UP TO START TABLE OVER!!
003634  017767  175476  175760  2$:     MOV     @$REGAD,PARITY          ;GET PARITY REGISTER TO BE USED
003642  162767  000002  175432          SUB     #2,$TSTNM               ;DECREASE TEST NO. FOR SCOPE
                                                                        ;LOOPING IN CASE TEST #4 WAS
                                                                        ;DONE BECAUSE TEST #4 BEING
                                                                        ;DONE THROWS $TSTNM COUNT OFF
                                                                        ;BY 2
003650  005737  002304                  TST     @#$KT11                 ;KT11 ARE YOU THERE?
003654  100006                          BPL     3$                      ;BRANCH IF NO
003656  017777  175534  174346          MOV     @$SETAD,@KPAR1          ;GET THE OFFSET NEEDED
003664  062767  000001  175410          ADD     #1,$TSTNM               ;SET TEST NO. TO A CORRECT
                                                                        ;VALUE, TEST #4 WAS NOT
                                                                        ;EXECUTED
003672  017700  175516          3$:     MOV     @$TMPAD,R0              ;GET PARITY MEMORY ASSOCIATED
```



```
                                                                        ;WITH THE PARITY REGISTER
003676  004337  003204                  JSR     R3,@#COMPUT             ;COMPUTE AN AREA IN THIS BANK
                                                                        ;FOR TESTING
003702  010137  001476                  MOV     R1,@#NEWSTK             ;SET UP A NEW STACK POINTER
                                                                        ;FOR STACK OPERATIONS IN CASE
                                                                        ;NO PARITY MEMORY RESIDES IN
                                                                        ;LOWER 4K
003706  004737  003376                  JSR     PC,@#WHICM1             ;DETERMINE IF WE ARE ABOUT TO
                                                                        ;TEST AN OLD MOS DESIGN!
003712  012705  011450                  MOV     #VECSET,R5              ;SET UP SERVICE ROUTINE ADDRESS
                                ;*************************************************************
                                ;TEST 6 TEST (ADDRESS)  SM0,DM3  MOV INSTRUCTION
                                ;*************************************************************
003716  000004                  TST6:   SCOPE
                                ;               11/45 **** ROM STATE 221 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
003720  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
003722  003752                          A                               ;ROUTINE ADDRESS
003724  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
003726  010010                          MOV     R0,@R0                  ;DO THE DATO
003730  012737  003740  001332          MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
003736  010030                          MOV     R0,@(R0)+               ;DO A DATI
003740  042777  000004  175654          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
003746  104001                          HLT     +1                      ;DIDN'T ABORT
003750  000410                          BR      .+22                    ;GO TO NEXT TEST
003752  042777  000005  175642  A1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
003760  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
003764  104003                          HLT     +3                      ;ABORTED INCORRECTLY
003766  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*************************************************************
                                ;TEST 7 TEST (ADDRESS)  SM0,DM5  MOV INSTRUCTION
                                ;*************************************************************
003772  000004                  TST7:   SCOPE
                                ;               11/45 **** ROM STATE 231 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
003774  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
003776  004026                          A0                              ;ROUTINE ADDRESS
004000  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004002  010020                          MOV     R0,(R0)+                ;DO THE DATO
004004  012737  004014  001332          MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004012  010050                          MOV     R0,@-(R0)               ;DO A DATI
004014  042777  000004  175600          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004022  104001                          HLT     +1                      ;DIDN'T ABORT
004024  000410                          BR      .+22                    ;GO TO NEXT TEST
004026  042777  000005  175566  A0:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004034  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004040  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004042  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
```

```
                                        ;*************************************************************
                                        ;TEST 10        TEST (DATA)  SM1,DM6  MOV INSTRUCTION
                                        ;*************************************************************
004046  000004                  TST10:  SCOPE
                                ;                       11/45 **** ROM STATE 27 ****
                                ;
                                ;       11/40 **** ROM STATE 206 ****
004050  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004052  004104                          A1                              ;ROUTINE ADDRESS
004054  011100                          MOV     #R1,R0                  ;SET UP FOR DATO
004056  010010                          MOV     R0,@R0                  ;DO THE DATO
004060  012737  004070  001332          MOV     #.+10,@#$GODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004066  011060  177776                  MOV     @R0,-2(R0)              ;DO A DATI
004072  042777  000004  175522          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004100  104001                          HLT     +1                      ;DIDN'T ABORT
004102  000410                          BR      .+22                    ;GO TO NEXT TEST
004104  042777  000005  175510  A1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004112  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004116  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004120  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                        ;*************************************************************
                                        ;TEST 11        TEST (ADDRESS)  SM0,DM7  MOV INSTRUCTION
                                        ;*************************************************************
004124  000004                  TST11:  SCOPE
                                ;                       11/45 **** ROM STATE 231 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
004126  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004130  004162                          A2                              ;ROUTINE ADDRESS
004132  011100                          MOV     #R1,R0                  ;SET UP FOR A DATO
004134  010020                          MOV     R0,(R0)+                ;DO THE DATO
004136  012737  004150  001332          MOV     #.+12,@#$GODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004144  010070  177776                  MOV     R0,@-2(R0)              ;DO A DATI
004150  042777  000004  175444          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004156  104001                          HLT     +1                      ;DIDN'T ABORT
004160  000410                          BR      .+22                    ;GO TO NEXT TEST
004162  042777  000005  175432  A2:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004170  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004174  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004176  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                        ;*************************************************************
                                        ;TEST 12        TEST (DATA)  SM0,DM2  CMP INSTRUCTION
                                        ;*************************************************************
004202  000004                  TST12:  SCOPE
                                ;                       11/45 **** ROM STATE 175 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004204  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004206  004236                          B                               ;ROUTINE ADDRESS
004210  011100                          MOV     #R1,R0                  ;SET UP FOR DATO
```



```
004212  010010                          MOV     R0,@R0                  ;DO THE DATO
004214  012737  004224  001332          MOV     #.+10,@#$GODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004222  020020                          CMP     R0,(R0)+                ;DO A DATIP, DATI
004224  042777  000004  175370          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004232  104001                          HLT     +1                      ;DIDN'T ABORT
004234  000410                          BR      .+22                    ;GO TO NEXT TEST
004236  042777  000005  175356  B1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004244  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004250  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004252  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                        ;*************************************************************
                                        ;TEST 13        TEST (DATA)  SM0,DM4  CMP INSTRUCTION
                                        ;*************************************************************
004256  000004                  TST13:  SCOPE
                                ;                       11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004260  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004262  004312                          B0                              ;ROUTINE ADDRESS
004264  011100                          MOV     #R1,R0                  ;SET UP FOR DATO
004266  010020                          MOV     R0,(R0)+                ;DO THE DATO
004270  012737  004300  001332          MOV     #.+10,@#$GODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004276  020040                          CMP     R0,-(R0)                ;DO A DATIP, DATI
004300  042777  000004  175314          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004306  104001                          HLT     +1                      ;DIDN'T ABORT
004310  000410                          BR      .+22                    ;GO TO NEXT TEST
004312  042777  000005  175302  B0:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004320  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004324  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004326  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                        ;*************************************************************
                                        ;TEST 14        TEST (DATA)  SM0,DM6  CMP INSTRUCTION
                                        ;*************************************************************
004332  000004                  TST14:  SCOPE
                                ;                       11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004334  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004336  004370                          B1                              ;ROUTINE ADDRESS
004340  011100                          MOV     #R1,R0                  ;SET UP FOR DATO
004342  010020                          MOV     R0,(R0)+                ;DO THE DATO
004344  012737  004356  001332          MOV     #.+12,@#$GODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004352  020060  177776                  CMP     R0,-2(R0)               ;DO A DATI, DATIP
004356  042777  000004  175236          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004364  104001                          HLT     +1                      ;DIDN'T ABORT
004366  000410                          BR      .+22                    ;GO TO NEXT TEST
004370  042777  000005  175224  B1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004376  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
```

```
1133  004402  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1134  004404  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1135                                          ;*********************************************************
1136                                          ;TEST 15        TEST (DATA)  SM0,DM1  CMP INSTRUCTION
1137                                          ;*********************************************************
1138  004410  000004                          TST15:  SCOPE
1139                                          ;               11/45 ***** ROM STATE 175 *****
1140                                          ;
1141                                          ;       11/40 **** ROM STATE 267 ****
1142  004412  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1143  004414  004444                          B2                              ;ROUTINE ADDRESS
1144  004416  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
1145  004420  010010                          MOV     R0,@R0                  ;DO THE DATO
1146  004422  012737  004432  001332          MOV     #.+10,@#SYDDAT          ;STORE THE PC THAT SHOULD
1147                                                                          ;BE PUSHED ON THE STACK
1148                                                                          ;IF A PARITY ABORT OCCURS
1149  004430  020010                          CMP     R0,@R0                  ;DO A DATIP, DATI
1150  004432  042777  000004  175162          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
1151  004440  104001                          HLT     +1                      ;DIDN'T ABORT
1152  004442  000410                          BR      .+22                    ;GO TO NEXT TEST
1153  004444  042777  000005  175150  B2:    BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1154  004452  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1155  004456  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1156  004460  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1157                                          ;*********************************************************
1158                                          ;TEST 16        TEST (DATA)  SM0,DM3  CMP INSTRUCTION
1159                                          ;*********************************************************
1160  004464  000004                          TST16:  SCOPE
1161                                          ;               11/45 ***** ROM STATE 177 *****
1162                                          ;
1163                                          ;       11/40 **** ROM STATE 267 ****
1164  004466  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1165  004470  004536                          B3                              ;ROUTINE ADDRESS
1166  004472  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
1167  004474  010020                          MOV     R0,(R0)+                ;DO THE DATO
1168  004476  042777  000004  175116          BIC     #BIT2,@PARITY           ;WRITE NORMAL
1169  004504  011110                          MOV     @R1,@R0                 ;WRITE ADDRESS NORMAL (DATI)
1170  004506  052777  000004  175106          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
1171  004514  012737  004524  001332          MOV     #.+10,@#SYDDAT          ;STORE THE PC THAT SHOULD
1172                                                                          ;BE PUSHED ON THE STACK
1173                                                                          ;IF A PARITY ABORT OCCURS
1174  004522  020030                          CMP     R0,@(R0)+               ;DO A DATI, DATIP
1175  004524  042777  000004  175070          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
1176  004532  104001                          HLT     +1                      ;DIDN'T ABORT
1177  004534  000410                          BR      .+22                    ;GO TO NEXT TEST
1178  004536  042777  000005  175056  B3:    BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1179  004544  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1180  004550  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1181  004552  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1182                                          ;*********************************************************
1183                                          ;TEST 17        TEST (ADDRESS)  SM0,DM3  CMP INSTRUCTION
1184                                          ;*********************************************************
1185  004556  000004                          TST17:  SCOPE
1186                                          ;               11/45 ***** ROM STATE 221 *****
```



```
1187                                          ;
1188                                          ;       11/40 **** ROM STATE 264 ****
1189  004560  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1190  004562  004612                          B4                              ;ROUTINE ADDRESS
1191  004564  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
1192  004566  010010                          MOV     R0,@R0                  ;DO THE DATO
1193  004570  012737  004600  001332          MOV     #.+10,@#SYDDAT          ;STORE THE PC THAT SHOULD
1194                                                                          ;BE PUSHED ON THE STACK
1195                                                                          ;IF A PARITY ABORT OCCURS
1196  004576  020030                          CMP     R0,@(R0)+               ;DO A DATI
1197  004600  042777  000004  175014          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
1198  004606  104001                          HLT     +1                      ;DIDN'T ABORT
1199  004610  000410                          BR      .+22                    ;GO TO NEXT TEST
1200  004612  042777  000005  175002  B4:    BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1201  004620  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1202  004624  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1203  004626  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1204                                          ;*********************************************************
1205                                          ;TEST 20        TEST DATI (DATA)  SM0,DM5  CMP INSTRUCTION
1206                                          ;*********************************************************
1207  004632  000004                          TST20:  SCOPE
1208                                          ;               11/45 ***** ROM STATE 177 *****
1209                                          ;
1210                                          ;       11/40 **** ROM STATE 267 ****
1211  004634  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1212  004636  004706                          B5                              ;ROUTINE ADDRESS
1213  004640  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
1214  004642  042777  000004  174752          BIC     #BIT2,@PARITY           ;WRITE NORMAL
1215  004650  010060  177776                  MOV     R0,-2(R0)               ;DO THE DATO
1216  004654  052777  000004  174740          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
1217  004662  011110                          MOV     @R1,@R0                 ;DO THE DATO
1218  004664  012737  004674  001332          MOV     #.+10,@#SYDDAT          ;STORE THE PC THAT SHOULD
1219                                                                          ;BE PUSHED ON THE STACK
1220                                                                          ;IF A PARITY ABORT OCCURS
1221  004672  020050                          CMP     R0,@-(R0)               ;DO A DATI, DATIP
1222  004674  042777  000004  174720          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
1223  004702  104001                          HLT     +1                      ;DIDN'T ABORT
1224  004704  000410                          BR      .+22                    ;GO TO NEXT TEST
1225  004706  042777  000005  174706  B5:    BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1226  004714  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1227  004720  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1228  004722  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1229                                          ;*********************************************************
1230                                          ;TEST 21        TEST (ADDRESS)  SM0,DM5  CMP INSTRUCTION
1231                                          ;*********************************************************
1232  004726  000004                          TST21:  SCOPE
1233                                          ;               11/45 ***** ROM STATE 231 *****
1234                                          ;
1235                                          ;       11/40 **** ROM STATE 264 ****
1236  004730  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1237  004732  004766                          B6                              ;ROUTINE ADDRESS
1238  004734  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
1239  004736  010010                          MOV     R0,(R0)                 ;DO THE DATO
1240  004740  062700  000002                  ADD     #2,R0
```

```
1241  004744  012737  004754  001332          MOV     #.+10,@#$$ODAT          ;STORE THE PC THAT SHOULD
1242                                                                          ;BE PUSHED ON THE STACK
1243                                                                          ;IF A PARITY ABORT OCCURS
1244  004752  020050                          CMP     R0,@-(R0)               ;DO A DATI
1245  004754  042777  000004  174640          BIC     #BIT2,@PAKITY           ;WRITE NORMAL FOR EMT CALL
1246  004762  104001                          HLT     +1                      ;DIDN'T ABORT
1247  004764  000410                          BR      .+22                    ;GO TO NEXT TEST
1248  004766  042777  000005  174626  B6:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1249  004774  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1250  005000  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1251  005002  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1252                                  ;*****************************************************************
1253                                  ;TEST 22        TEST (DATA)  SM2,DM0  CMP INSTRUCTION
1254                                  ;*****************************************************************
1255  005006  000004                  TST22:  SCOPE
1256                                  ;                       11/45 **** ROM STATE 27 ****
1257                                  ;
1258                                  ;       11/40 **** ROM STATE 250 ****
1259  005010  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1260  005012  005042                          C                               ;ROUTINE ADDRESS
1261  005014  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
1262  005016  010010                          MOV     R0,@R0                  ;DO THE DATO
1263  005020  012737  005030  001332          MOV     #.+10,@#$$ODAT          ;STORE THE PC THAT SHOULD
1264                                                                          ;BE PUSHED ON THE STACK
1265                                                                          ;IF A PARITY ABORT OCCURS
1266  005026  022000                          CMP     (R0)+,R0                ;DO A DATI
1267  005030  042777  000004  174564          BIC     #BIT2,@PAKITY           ;WRITE NORMAL FOR EMT CALL
1268  005036  104001                          HLT     +1                      ;DIDN'T ABORT
1269  005040  000410                          BR      .+22                    ;GO TO NEXT TEST
1270  005042  042777  000005  174552  C:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1271  005050  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1272  005054  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1273  005056  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1274                                  ;*****************************************************************
1275                                  ;TEST 23        TEST (DATA)  SM4,DM0  CMP INSTRUCTION
1276                                  ;*****************************************************************
1277  005062  000004                  TST23:  SCOPE
```



```
1278                                  ;                       11/45 **** ROM STATE 27 ****
```

```
1279                                    ;
1280                                    ;       11/40 **** ROM STATE 250 ****
1281  005064  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1282  005066  005116                            C0                              ;ROUTINE ADDRESS
1283  005070  011100                            MOV     @R1,R0                  ;SET UP FOR DATO
1284  005072  010020                            MOV     R0,(R0)+                ;DO THE DATO
```



```
1285  005074  012737  005104  001332            MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
1286                                                                            ;BE PUSHED ON THE STACK
1287                                                                            ;IF A PARITY ABORT OCCURS
1288  005102  024000                            CMP     -(R0),R0                ;DO A DATI
1289  005104  042777  000004  174510            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
1290  005112  104001                            HLT     +1                      ;DIDN'T ABORT
1291  005114  000410                            BR      .+22                    ;GO TO NEXT TEST
1292  005116  042777  000005  174476  C0:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1293  005124  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1294  005130  104003                            HLT     +3                      ;ABORTED INCORRECTLY
1295  005132  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
1296                                    ;*******************************************************
1297                                    ;TEST 24        TEST (DATA)  SM3,DM0  CMP INSTRUCTION
1298                                    ;*******************************************************
1299  005136  000004                    TST24:  SCOPE
1300                                    ;                 11/45 **** ROM STATE 146 ****
1301                                    ;
1302                                    ;       11/40 **** ROM STATE 250 ****
1303  005140  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1304  005142  005210                            C1                              ;ROUTINE ADDRESS
1305  005144  011100                            MOV     @R1,R0                  ;SET UP FOR DATO
1306  005146  010020                            MOV     R0,(R0)+                ;DO THE DATO (DATA OTHER PARITY)
1307  005150  042777  000004  174444            BIC     #BIT2,@PARITY           ;WRITE NORMAL
1308  005156  011110                            MOV     @R1,@R0                 ;DO A DATO (ADDRESS NORMAL)
1309  005160  052777  000004  174434            BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
1310  005166  012737  005176  001332            MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
1311                                                                            ;BE PUSHED ON THE STACK
1312                                                                            ;IF A PARITY ABORT OCCURS
1313  005174  023000                            CMP     @(R0)+,R0               ;DO A DATI
1314  005176  042777  000004  174416            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
1315  005204  104001                            HLT     +1                      ;DIDN'T ABORT
1316  005206  000410                            BR      .+22                    ;GO TO NEXT TEST
1317  005210  042777  000005  174404  C1:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1318  005216  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1319  005222  104003                            HLT     +3                      ;ABORTED INCORRECTLY
1320  005224  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
1321                                    ;*******************************************************
1322                                    ;TEST 25        TEST (DATA)  SM5,DM0  CMP INSTRUCTION
1323                                    ;*******************************************************
1324  005230  000004                    TST25:  SCOPE
1325                                    ;                 11/45 **** ROM STATE 146 ****
1326                                    ;
1327                                    ;       11/40 **** ROM STATE 250 ****
1328  005232  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1329  005234  005304                            C2                              ;ROUTINE ADDRESS
1330  005236  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
1331  005240  042777  000004  174354            BIC     #BIT2,@PARITY           ;WRITE NORMAL
1332  005246  010060  177776                    MOV     R0,-2(R0)               ;DO A DATO (ADDRESS NORMAL)
1333  005252  052777  000004  174342            BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
1334  005260  011110                            MOV     @R1,@R0                 ;DO A DATO (DATA OTHER PARITY)
1335  005262  012737  005272  001332            MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
1336                                                                            ;BE PUSHED ON THE STACK
1337                                                                            ;IF A PARITY ABORT OCCURS
1338  005270  025000                            CMP     @-(R0),R0               ;DO A DATI
```

```
1339  005272  042777  000004  174322          BIC     #BIT2,@PAMITY           ;WRITE NORMAL FOR EMT CALL
1340  005300  104001                          HLT     +1                      ;DIDN'T ABORT
1341  005302  000410                          BR      .+22                    ;GO TO NEXT TEST
1342  005304  042777  000005  174310  C4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1343  005312  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1344  005316  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1345  005320  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1346                                  ;*************************************************************
1347                                  ;TEST 26        TEST (DATA)  SM1,DM0  CMP INSTRUCTION
1348                                  ;*************************************************************
1349  005324  000004                  TST26:  SCOPE
1350                                  ;                       11/45 ***** ROM STATE 27 ****
1351                                  ;
1352                                  ;       11/40 **** ROM STATE 250 ****
1353  005326  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1354  005330  005360                          C3                              ;ROUTINE ADDRESS
1355  005332  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
1356  005334  010010                          MOV     R0,@R0                  ;DO THE DATO
1357  005336  012737  005346  001332          MOV     #.+10,@#$ODDAT          ;STORE THE PC THAT SHOULD
1358                                                                          ;BE PUSHED ON THE STACK
1359                                                                          ;IF A PARITY ABORT OCCURS
1360  005344  021000                          CMP     @R0,R0                  ;DO A DATI
1361  005346  042777  000004  174246          BIC     #BIT2,@PAMITY           ;WRITE NORMAL FOR EMT CALL
1362  005354  104001                          HLT     +1                      ;DIDN'T ABORT
1363  005356  000410                          BR      .+22                    ;GO TO NEXT TEST
1364  005360  042777  000005  174234  C5:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1365  005366  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1366  005372  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1367  005374  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1368                                  ;*************************************************************
1369                                  ;TEST 27        TEST (DATA)  SM6,DM0  CMP INSTRUCTION
1370                                  ;*************************************************************
1371  005400  000004                  TST27:  SCOPE
1372                                  ;                       11/45 ***** ROM STATE 142 ****
1373                                  ;
1374                                  ;       11/40 **** ROM STATE 250 ****
1375  005402  004015                          JSR R0,(R5)                     ;SET UP PARITY VECTOR SERVICE
1376  005404  005436                          C4                              ;ROUTINE ADDRESS
1377  005406  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
1378  005410  010020                          MOV     R0,(R0)+                ;DO THE DATO
1379  005412  012737  005424  001332          MOV     #.+12,@#$ODDAT          ;STORE THE PC THAT SHOULD
1380                                                                          ;BE PUSHED ON THE STACK
1381                                                                          ;IF A PARITY ABORT OCCURS
1382  005420  026000  177776                  CMP     -2(R0),R0               ;DO A DATI
1383  005424  042777  000004  174170          BIC     #BIT2,@PAMITY           ;WRITE NORMAL FOR EMT CALL
1384  005432  104001                          HLT     +1                      ;DIDN'T ABORT
1385  005434  000410                          BR      .+22                    ;GO TO NEXT TEST
1386  005436  042777  000005  174156  C4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1387  005444  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1388  005450  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1389  005452  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1390                                  ;*************************************************************
1391                                  ;TEST 30        TEST (ADDRESS)  SM7,DM0  CMP INSTRUCTION
1392                                  ;*************************************************************
```



```
1393  005456  000004                  TST30:  SCOPE
1394                                  ;                       11/45 ***** ROM STATE 142 ****
1395                                  ;
1396                                  ;       11/40 **** ROM STATE 245 ****
1397  005460  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1398  005462  005514                          C5                              ;ROUTINE ADDRESS
1399  005464  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
1400  005466  010020                          MOV     R0,(R0)+                ;DO THE DATO
1401  005470  012737  005502  001332          MOV     #.+12,@#$ODDAT          ;STORE THE PC THAT SHOULD
1402                                                                          ;BE PUSHED ON THE STACK
1403                                                                          ;IF A PARITY ABORT OCCURS
1404  005476  027000  177776                  CMP     @-2(R0),R0              ;DO A DATI
1405  005502  042777  000004  174112          BIC     #BIT2,@PAMITY           ;WRITE NORMAL FOR EMT CALL
1406  005510  104001                          HLT     +1                      ;DIDN'T ABORT
1407  005512  000410                          BR      .+22                    ;GO TO NEXT TEST
1408  005514  042777  000005  174100  C5:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1409  005522  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1410  005526  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1411  005530  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1412                                  ;*************************************************************
1413                                  ;TEST 31        TEST (ADDRESS)  SM3,DM0  CMP INSTRUCTION
1414                                  ;*************************************************************
1415  005534  000004                  TST31:  SCOPE
1416                                  ;                       11/45 ***** ROM STATE 27 ****
1417                                  ;
1418                                  ;       11/40 **** ROM STATE 245 ****
1419  005536  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1420  005540  005570                          C6                              ;ROUTINE ADDRESS
1421  005542  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
1422  005544  010010                          MOV     R0,@R0                  ;DO THE DATO
1423  005546  012737  005556  001332          MOV     #.+10,@#$ODDAT          ;STORE THE PC THAT SHOULD
1424                                                                          ;BE PUSHED ON THE STACK
1425                                                                          ;IF A PARITY ABORT OCCURS
1426  005554  023000                          CMP     @(R0)+,R0               ;DO A DATI
1427  005556  042777  000004  174036          BIC     #BIT2,@PAMITY           ;WRITE NORMAL FOR EMT CALL
1428  005564  104001                          HLT     +1                      ;DIDN'T ABORT
1429  005566  000410                          BR      .+22                    ;GO TO NEXT TEST
1430  005570  042777  000005  174024  C6:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1431  005576  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1432  005602  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1433  005604  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
1434                                  ;*************************************************************
1435                                  ;TEST 32        TEST (ADDRESS)  SM5,DM0  CMP INSTRUCTION
1436                                  ;*************************************************************
1437  005610  000004                  TST32:  SCOPE
1438                                  ;                       11/45 ***** ROM STATE 27 ****
1439                                  ;
1440                                  ;       11/40 **** ROM STATE 245 ****
1441  005612  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1442  005614  005644                          C7                              ;ROUTINE ADDRESS
1443  005616  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
1444  005620  010020                          MOV     R0,(R0)+                ;DO THE DATO
1445  005622  012737  005632  001332          MOV     #.+10,@#$ODDAT          ;STORE THE PC THAT SHOULD
1446                                                                          ;BE PUSHED ON THE STACK
```

```
1447                                                                    ;IF A PARITY ABORT OCCURS
1448  005630  025000                            CMP     #-(R0),R0       ;DO A DATI
1449  005632  042777  000004  173762            BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
1450  005640  104001                            HLT     +1              ;DIDN'T ABORT
1451  005642  000410                            BR      .+22            ;GO TO NEXT TEST
1452  005644  042777  000005  173750  C7:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1453  005652  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
1454  005656  104003                            HLT     +3              ;ABORTED INCORRECTLY
1455  005660  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
1456                                  ;*********************************************************
1457                                  ;TEST 33        TEST (DATA)  SM7,DM0  CMP INSTRUCTION
1458                                  ;*********************************************************
1459  005664  000004                TST33:  SCOPE
1460                                  ;                       11/45 ***** ROM STATE 146 ****
1461                                  ;
1462                                  ;       11/40 **** ROM STATE 250 ****
1463  005666  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
1464  005670  005742                            C8                      ;ROUTINE ADDRESS
1465  005672  011100                            MOV     @R1,R0          ;SET UP FOR A DATO
1466  005674  042777  000004  173720            BIC     #BIT2,@PARITY   ;WRITE NORMAL
1467  005702  010060  177776                    MOV     R0,-2(R0)       ;DO A DATO (ADDRESS NORMAL)
1468  005706  052777  000004  173706            BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
1469  005714  011110                            MOV     @R1,@R0         ;DO A DATO (DATA OTHER PARITY)
1470  005716  012737  005730  001332            MOV     #.+12,@#$DDAT   ;STORE THE PC THAT SHOULD
1471                                                                    ;BE PUSHED ON THE STACK
1472                                                                    ;IF A PARITY ABORT OCCURS
1473  005724  047000  177776                    CMP     @-2(R0),R0      ;DO A DATI
1474  005730  042777  000004  173664            BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
1475  005736  104001                            HLT     +1              ;DIDN'T ABORT
1476  005740  000410                            BR      .+22            ;GO TO NEXT TEST
1477  005742  042777  000005  173652  C8:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1478  005750  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
1479  005754  104003                            HLT     +3              ;ABORTED INCORRECTLY
1480  005756  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
1481                                  ;*********************************************************
1482                                  ;TEST 34        TEST  DM3  JMP INSTRUCTION
1483                                  ;*********************************************************
1484  005762  000004                TST34:  SCOPE
1485                                  ;                       11/45 ***** ROM STATE 221 ****
1486                                  ;
1487                                  ;       11/40 **** ROM STATE 303 ****
1488  005764  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
1489  005766  006020                            D                       ;ROUTINE ADDRESS
1490  005770  011100                            MOV     @R1,R0          ;SET UP FOR A DATO
1491  005772  012710  006006                    MOV     #DD,@R0         ;DO THE DATO
1492  005776  012737  006006  001332            MOV     #.+10,@#$DDAT   ;STORE THE PC THAT SHOULD
1493                                                                    ;BE PUSHED ON THE STACK
1494                                                                    ;IF A PARITY ABORT OCCURS
1495  006004  000130                            JMP     @(R0)+          ;DO A DATI
1496  006006  042777  000004  173606  DD:       BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
1497  006014  104001                            HLT     +1              ;DIDN'T ABORT
1498  006016  000410                            BR      .+22            ;GO TO NEXT TEST
1499  006020  042777  000005  173574  D8:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1500  006026  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
```



```
1501  006032  104003                            HLT     +3              ;ABORTED INCORRECTLY
1502  006034  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
1503                                  ;*********************************************************
1504                                  ;TEST 35        TEST  DM5  JMP INSTRUCTION
1505                                  ;*********************************************************
1506  006040  000004                TST35:  SCOPE
1507                                  ;                       11/45 ***** ROM STATE 231 ****
1508                                  ;
1509                                  ;       11/40 **** ROM STATE 303 ****
1510  006042  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
1511  006044  006076                            D0                      ;ROUTINE ADDRESS
1512  006046  011100                            MOV     @R1,R0          ;SET UP FOR A DATO
1513  006050  012720  006064                    MOV     #DD0,(R0)+      ;DO THE DATO
1514  006054  012737  006064  001332            MOV     #.+10,@#$DDAT   ;STORE THE PC THAT SHOULD
1515                                                                    ;BE PUSHED ON THE STACK
1516                                                                    ;IF A PARITY ABORT OCCURS
1517  006062  000150                            JMP     @-(R0)          ;DO A DATI
1518  006064  042777  000004  173530  DD0:      BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
1519  006072  104001                            HLT     +1              ;DIDN'T ABORT
1520  006074  000410                            BR      .+22            ;GO TO NEXT TEST
1521  006076  042777  000005  173516  D0:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1522  006104  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
1523  006110  104003                            HLT     +3              ;ABORTED INCORRECTLY
1524  006112  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
1525                                  ;*********************************************************
1526                                  ;TEST 36        TEST  DM7  JMP INSTRUCTION
1527                                  ;*********************************************************
1528  006116  000004                TST36:  SCOPE
1529                                  ;                       11/45 ***** ROM STATE 231 ****
1530                                  ;
1531                                  ;       11/40 **** ROM STATE 303 ****
1532  006120  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
1533  006122  006156                            D1                      ;ROUTINE ADDRESS
1534  006124  011100                            MOV     @R1,R0          ;SET UP FOR A DATO
1535  006126  012720  006144                    MOV     #DD1,(R0)+      ;DO THE DATO
1536  006132  012737  006144  001332            MOV     #.+12,@#$DDAT   ;STORE THE PC THAT SHOULD
1537                                                                    ;BE PUSHED ON THE STACK
1538                                                                    ;IF A PARITY ABORT OCCURS
1539  006140  000170  177776                    JMP     @-2(R0)         ;DO A DATI
1540  006144  042777  000004  173450  DD1:      BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
1541  006152  104001                            HLT     +1              ;DIDN'T ABORT
1542  006154  000410                            BR      .+22            ;GO TO NEXT TEST
1543  006156  042777  000005  173436  D1:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1544  006164  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
1545  006170  104003                            HLT     +3              ;ABORTED INCORRECTLY
1546  006172  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
1547
1548
        ;*********************************************************
        ;*********************************************************
        ;*********************************************************
        ;*********************************************************
        ;
        ;
```

```
                        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
                        ;
                        ;1ST PUSH - ADDRESS OF THE TAG "VECSET" (NORMAL)
                        ;           THIS ADDRESS WOULD BE PLACED IN
                        ;           R5 UPON COMPLETION OF "RTS R5"
                        ;           INSTRUCTION
                        ;2ND PUSH
                        ;   .
                        ;   .           NO. OF PARAMETERS AS A FUNCTION
                        ;   .           OF MEMORY INTERLEAVING
                        ;   .
                        ;NTH PUSH
                        ;NTH +1 PUSH - MARK INSTRUCTION (OTHER PARITY)
                        ;LAST PUSH - OLD PC FROM THE "JSR" (NORMAL)
                        ;
                        ;NOTE:  THE TEST SHOULD ABORT ON ATTEMPT TO FETCH THE
                        ;       MARK INSTRUCTION (NTH +1 PUSH)
                        ;
                        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
                        ;       POSITIONED AT THE NTH +1 PUSH, THUS GIVING,
                        ;
                        ;NTH +2 PUSH - PS FROM PARITY ERROR
                        ;NTH +3 PUSH - PC FROM PARITY ERROR
                        ;
                        ;
                        ;*********************************************************
                        ;*********************************************************
                        ;*********************************************************
                        ;*********************************************************
 1584
 1585
 1586
 1587                                   ;*****************************************************
 1588                                   ;TEST 37        TEST   MARK "POP"   MARK INSTRUCTION
 1589                                   ;*****************************************************
 1590  006176  000004                  TST37:  SCOPE
 1591                                   ;                       11/45 ***** ROM STATE 260 ****
 1592                                   ;
 1593                                   ;       11/40 **** ROM STATE 1 ****
 1594  006200  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 1595  006202  006326                          E0                              ;ROUTINE ADDRESS
 1596  006204  042777  000004  173410          BIC     #BIT2,@PAKITY           ;WRITE NORMAL
 1597                                   ;*****************************************************
 1598                                   ;*****************************************************
 1599                                   ;*****************************************************
 1600                                   ;*****************************************************
 1601                                   ;
 1602                                   ;
 1603                                   ;NOTE THAT THE NEXT INSTRUCTION WILL MOVE THE STACK TO PARITY AREA
 1604                                   ;IN THE EVENT THAT NO PARITY EXISTS WHERE THE STACK IS NORMALLY
 1605                                   ;SITTING AT 1100
 1606                                   ;
 1607                                   ;
 1608                                   ;*****************************************************
```



```
 1609                                   ;*****************************************************
 1610                                   ;*****************************************************
 1611                                   ;*****************************************************
 1612  006212  013706  001476                  MOV     @#NEWSTK,SP             ;SET THE STACK IN PARITY
 1613                                                                           ;MEMORY AREA
 1614  006216  017702  173224                  MOV     @INTERAD,R2             ;GET THE INTERLEAVE FACTOR
 1615                                                                           ;FOR THIS CONTROLER
 1616  006222  005302                          DEC     R2                      ;CALCULATE NO. OF PARAMETERS
 1617                                                                           ;TO BE PUSHED ON THE STACK
 1618  006224  010546                          MOV     R5,-(SP)                ;PUSH R5 CONTENTS ON STACK
 1619  006226  005702                  1$:     TST     R2                      ;ANY PARAMETERS TO BE
 1620                                                                           ;PUSHED ON THE STACK?
 1621  006230  001404                          BEQ     2$                      ;BRANCH IF NO
 1622  006232  005302                          DEC     R2                      ;SUBTRACT 1 FROM PARAMETER COUNT
 1623  006234  012746  000001                  MOV     #1,-(SP)                ;PUSH PARAMETER ONTO STACK
 1624  006240  000772                          BR      1$                      ;GO BACK TO SEE IF ANY MORE!
 1625  006242  052777  000004  173352  2$:     BIS     #BIT2,@PAKITY           ;WRITE OTHER PARITY
 1626  006250  017702  173172                  MOV     @INTERAD,R2             ;GET THE INTERLEAVE FACTOR
 1627                                                                           ;FOR THIS CONTROLLER
 1628  006254  005302                          DEC     R2                      ;CALCULATE NO. OF PARAMETERS
 1629                                                                           ;THAT WERE TO BE PUSHED ON THE
 1630                                                                           ;STACK
 1631  006256  062702  006400                  ADD     #6400,R2                ;CALCULATE THE CORRESPONDING
 1632                                                                           ;MARK INSTRUCTION
 1633  006262  010246                          MOV     R2,-(SP)                ;PUSH MARK INSTRUCTION ON STACK
 1634  006264  042777  000004  173330          BIC     #BIT2,@PAKITY           ;WRITE NORMAL
 1635  006272  010605                          MOV     SP,R5                   ;PLACE MARK INSTRUCTION ADDRESS
 1636                                                                           ;INTO R5 FOR RTS
 1637  006274  010602                          MOV     SP,R2                   ;GET THE PC OF THE MARK
 1638                                                                           ;INSTRUCTION
 1639  006276  005737  001642                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
 1640  006302  001002                          BNE     3$                      ;BRANCH IF YES
 1641                                                                           ;AND DON'T STEP UP THE PC
 1642                                                                           ;IT WON'T BE UPDATED ON THE
 1643                                                                           ;PARITY ABORT
 1644  006304  062702  000002                  ADD     #2,R2                   ;STEP UP THE PC
 1645                                                                           ;IT WILL BE UPDATED ON THE
 1646                                                                           ;PARITY ABORT
 1647  006310  010237  001332          3$:     MOV     R2,@#$GOODAT            ;STORE THIS VALUE OF PC
 1648                                                                           ;THAT SHOULD BE PUSHED ON THE
 1649                                                                           ;STACK IF A PARITY ABORT OCCURS
 1650  006314  004767  000004                  JSR     PC,MRK0                 ;SUBROUTINE CALL
 1651  006320  104001                  E1:     HLT     +1                      ;DIDN'T ABORT
 1652  006322  000407                          BR      .+20                    ;GO TO RESET THE STACK
 1653  006324  000205                  MRK0:   RTS     R5                      ;RETURN FROM SUBROUTINE
 1654  006326  042777  000001  173266  E0:     BIC     #BIT0,@PAKITY           ;DISABLE PARITY
 1655  006334  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 1656  006340  104003                          HLT     +3                      ;ABORTED INCORRECTLY
 1657  006342  013706  001476                  MOV     @#NEWSTK,SP             ;RESET THE STACK
 1658
 1659
 1660
                        ;*********************************************************
                        ;*********************************************************
```

```
        ;********************************************************************
        ;********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS
        ;
        ;1ST PUSH = ADDRESS OF THE TAG 'VECSET' (OTHER PARITY)
        ;          THIS ADDRESS WOULD BE PLACED IN
        ;          R5 UPON COMPLETION OF 'RTS R5'
        ;2ND PUSH
        ;   .
        ;   .          NO. OF PARAMETERS AS A FUNCTION
        ;   .          OF MEMORY INTERLEAVING
        ;   .
        ;NTH PUSH
        ;NTH +1 PUSH = MARK INSTRUCTION  (NORMAL)
        ;LAST PUSH = OLD PC FROM THE 'JSR' (NORMAL)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
        ;       R5 CONTENTS (1ST PUSH)
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
        ;       PROPERLY UPDATED, THUS GIVING,
        ;
        ;1ST PUSH = PS FROM THE PARITY ERROR
        ;2ND PUSH = PC FROM THE PARITY ERROR
        ;
        ;
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
 1695
 1696
 1697
 1698  006346  012705  011450                MOV     #VECSET,R5              ;RESTORE THE PARITY VECTOR
 1699                                                                          ;SERVICE ADDRESS SETUP ROUTINE
 1700                                                                          ;ADDRESS
 1701                                ;*************************************************************
 1702                                ;TEST 40        TEST   OLD REGISTER CONTENTS   MARK INSTRUCTION
 1703                                ;*************************************************************
 1704  006352  000004                TST40:  SCOPE
 1705                                ;                       11/45 **** ROM STATE 235 ****
 1706                                ;
 1707                                ;       11/40 **** ROM STATE 356 ****
 1708  006354  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 1709  006356  006510                        E2                              ;ROUTINE ADDRESS
 1710  006360  017702  173062                MOV     @NTERAD,R2              ;GET THE INTERLEAVE FACTOR
 1711                                                                          ;FOR THIS CONTROLLER
 1712  006364  005302                        DEC     R2                      ;CALCULATE NO. OF PARAMETERS
 1713                                                                          ;TO BE PUSHED ON THE STACK
 1714  006366  042777  000004  173226        BIC     #BIT2,@PANITY           ;WRITE NORMAL
 1715  006374  005737  001642                TST     @#CPU40                 ;ARE WE ON AN 11/40?
 1716  006400  001407                        BEQ     3$                      ;BRANCH IF NO
```



```
 1717                                                                          ;DATA WILL BE TAKEN FIRST AND
 1718                                                                          ;THEN THE PC UPDATED
 1719  006402  013700  001476                MOV     @#NEWSTK,R0             ;GET THE INITIAL STACK POINT
 1720  006406  162700  000002                SUB     #2,R0                   ;DROP DOWN 1 WORD ADDRESS
 1721                                                                          ;BECAUSE THE PC WILL BE UPDATED
 1722                                                                          ;FIRST THEN THE DATA TAKEN
 1723  006412  010037  001332                MOV     R0,@#$GDDAT             ;STORE THIS VALUE AS THE PC
 1724                                                                          ;THAT SHOULD BE PUSHED ON THE STACK
 1725                                                                          ;IF A PARITY ABORT OCCURS
 1726  006416  000403                        BR      4$                      ;CONTINUE WITH TEST
 1727  006420  012737  006502  001332  3$:   MOV     #E1,@#$GDDAT            ;STORE THE PC THAT SHOULD
 1728                                                                          ;BE PUSHED ON THE STACK
 1729                                                                          ;IF A PARITY ABORT OCCURS
 1730  006426  052777  000004  173166  4$:   BIS     #BIT2,@PANITY           ;WRITE OTHER PARITY
 1731  006434  010546                        MOV     R5,-(SP)                ;STORE OLD R5 CONTENTS ON STACK
 1732  006436  042777  000004  173156        BIC     #BIT2,@PANITY           ;WRITE NORMAL
 1733  006444  005702                1$:     TST     R2                      ;ANY PARAMETERS TO BE PUSHED
 1734                                                                          ;ON THE STACK?
 1735  006446  001404                        BEQ     2$                      ;BRANCH IF NO
 1736  006450  005302                        DEC     R2                      ;SUBTRACT 1 FROM PARAMETER COUNT
 1737  006452  012746  000001                MOV     #1,-(SP)                ;PUSH PARAMETER ON STACK
 1738  006456  000772                        BR      1$                      ;GO BACK TO SEE IF ANY MORE!
 1739  006460  017702  172762          2$:   MOV     @NTERAD,R2              ;GET THE INTERLEAVE FACTOR
 1740                                                                          ;FOR THIS CONTROLLER
 1741  006464  005302                        DEC     R2                      ;CALCULATE NO. OF PARAMETERS
 1742                                                                          ;THAT WERE TO BE PUSHED ON THE
 1743                                                                          ;STACK
 1744  006466  062702  006400                ADD     #6400,R2                ;CALCULATE THE CORRESPONDING
 1745                                                                          ;MARK INSTRUCTION
 1746  006472  010246                        MOV     R2,-(SP)                ;PUSH MARK INSTRUCTION ON STACK
 1747  006474  010605                        MOV     SP,R5                   ;PLACE MARK INSTRUCTION ADDRESS
 1748                                                                          ;INTO R5 FOR RTS
 1749  006476  004767  000004                JSR     PC,MRK1                 ;SUBROUTINE CALL
 1750  006502  104001                E1:     HLT     +1                      ;DIDN'T ABORT
 1751  006504  000407                        BR      .+20                    ;GO TO RESET THE STACK
 1752  006506  000205                MRK1:   RTS     R5                      ;RETURN FROM SUBROUTINE
 1753  006510  042777  000001  173104  E2:   BIC     #BIT0,@PANITY           ;DISABLE PARITY
 1754  006516  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 1755  006522  104003                        HLT     +3                      ;ABORTED INCORRECTLY
 1756  006524  013706  001476                MOV     @#NEWSTK,SP             ;RESET THE STACK
 1757  006530  012705  011450                MOV     #VECSET,R5              ;RESTORE THE PARITY VECTOR
 1758                                                                          ;SERVICE ADDRESS SETUP ROUTINE
 1759                                                                          ;ADDRESS
 1760                                ;*************************************************************
 1761                                ;TEST 41        TEST   SOB BRANCH   SOB INSTRUCTION
 1762                                ;*************************************************************
 1763  006534  000004                TST41:  SCOPE
 1764                                ;                       11/45 **** ROM STATE 260 ****
 1765                                ;
 1766                                ;       11/40 **** ROM STATE 1 ****
 1767  006536  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 1768  006540  006644                        F0                              ;ROUTINE ADDRESS
 1769  006542  042777  000004  173052        BIC     #BIT2,@PANITY           ;WRITE NORMAL
 1770  006550  012700  000004                MOV     #4,R0                   ;MOVE A NUMBER >1 TO R0 SO THAT
```

```
1771                                                                    ;WHEN DECREMENTED BY "SOB" THE
1772                                                                    ;RESULT WON'T BE 0
1773  006554  011102                          MOV     #R1,R2            ;SET UP FOR A DATO
1774  006556  013703  001476                  MOV     @#NEWSTK,R3       ;GET THE INITIAL TEST POINT
1775  006562  005737  001642                  TST     @#CPU40           ;ARE WE ON AN 11/40?
1776  006566  001403                          BEQ     1$                ;BRANCH IF NO
1777  006570  062703  000002                  ADD     #2,R3             ;STEP UP 1 WORD
1778                                                                    ;SINCE THE PC WILL NOT BE
1779                                                                    ;UPDATED ON THE PARITY ABORT
1780  006574  000402                          BR      2$                ;GO TO STORE THIS VALUE
1781  006576  062703  000004          1$:     ADD     #4,R3             ;STEP UP 2 WORDS
1782                                                                    ;SINCE THE PC WILL BE UPDATED
1783                                                                    ;ON THE PARITY ABORT
1784  006602  010337  001332          2$:     MOV     R3,@#$GOODAT      ;STORE THIS VALUE OF PC
1785                                                                    ;THAT SHOULD BE PUSHED ON THE
1786                                                                    ;STACK IF A PARITY ABORT OCCURS
1787  006606  052777  000004  173006          BIS     #BIT2,@PARITY     ;WRITE OTHER PARITY
1788  006614  012722  000240                  MOV     #240,(R2)+        ;INSTRUCTION TO BE DONE ON
1789                                                                    ;COMPLETION OF SOB INSTRUCTION
1790                                                                    ;THE INSTRUCTION WILL BE IN
1791                                                                    ;PARITY MEMORY AS DESIGNATED BY
1792                                                                    ;THE CONTENTS OF R1
1793  006620  042777  000004  172774          BIC     #BIT2,@PARITY     ;WRITE NORMAL
1794  006626  012712  000203                  MOV     #203,@R2          ;MOVE AN RTS R3 INTO THE LOCATION
1795                                                                    ;FOLLOWING THE INSTRUCTION TO BE
1796                                                                    ;EXECUTED IN PARITY MEMORY IN
1797                                                                    ;CASE IT DOESN'T ABORT
1798  006632  000403                          BR      .+10              ;GO TO SOB INSTRUCTION
1799  006634  004342                  F1:     JSR     R3,-(R2)          ;THE INSTRUCTION TO BE DONE ON
1800                                                                    ;COMPLETION OF SOB EXECUTION IS
1801                                                                    ;PLACED HERE
1802  006636  104001                          HLT     +1                ;DIDN'T ABORT
1803  006640  000407                          BR      .+20              ;GO TO RESET THE STACK
1804  006642  077004                          SOB     R0,F              ;EXECUTE THIS INSTRUCTION
1805  006644  042777  000001  172750  F0:     BIC     #BIT0,@PARITY     ;DISABLE PARITY
1806  006652  004037  011550                  JSR     R0,@#CHECKLOC     ;CHECK FOR GOOD ABORT
1807  006656  104003                          HLT     +3                ;ABORTED INCORRECTLY
1808  006660  013706  001476                  MOV     @#NEWSTK,SP       ;RESET THE STACK
1809                                  ;*********************************************************
1810                                  ;TEST 42        TEST  SM0,DM0  MOV INSTRUCTION
1811                                  ;*********************************************************
1812  006664  000004                  TST42:  SCOPE
1813                                  ;                   11/45 **** ROM STATE 20 ****
1814                                  ;
1815                                  ;       11/40 **** ROM STATE 1 ****
1816  006666  004015                          JSR     R0,(R5)           ;SET UP PARITY VECTOR SERVICE
1817  006670  006774                          F1                        ;ROUTINE ADDRESS
1818  006672  011100                          MOV     @R1,R0            ;SET UP FOR A DATO
1819  006674  017702  172546                  MOV     @INTERAD,R2       ;GET THE INTERLEAVE FACTOR
1820                                                                    ;FOR THIS CONTROLLER
1821  006700  042777  000004  172714          BIC     #BIT2,@PARITY     ;WRITE NORMAL
1822  006706  005702                  1$:     TST     R2                ;ANY 'MOV R0,R0'S TO BE MOVED
1823                                                                    ;INTO PARITY MEMORY AREA?
1824  006710  001404                          BEQ     2$                ;BRANCH IF NO
```



```
1825  006712  005302                          DEC     R2                ;SUBTRACT 1 FROM INSTRUCTION
1826                                                                    ;COUNT
1827  006714  012720  010000                  MOV     #010000,(R0)+     ;MOVE THE INSTRUCTION 'MOV R0,R0'
1828                                                                    ;TO PARITY AREA
1829  006720  000772                          BR      1$                ;GO BACK TO SEE IF ANY MORE!
1830  006722  052777  000004  172672  2$:     BIS     #BIT2,@PARITY     ;WRITE OTHER PARITY
1831  006730  012720  010000                  MOV     #010000,(R0)+     ;MOVE THE INSTRUCTION 'MOV R0,R0'
1832                                                                    ;INTO THE NEXT WORD LOCATION AFTER
1833                                                                    ;THE PREVIOUS 'MOV R0,R0' INSTRUCTION
1834  006734  042777  000004  172660          BIC     #BIT2,@PARITY     ;WRITE NORMAL
1835  006742  012710  000203                  MOV     #203,(R0)         ;MOVE 'RTS R3' INTO NEXT WORD
1836                                                                    ;LOCATION AFTER THE PREVIOUS
1837                                                                    ;'MOV R0,R0' INSTRUCTION IN CASE
1838                                                                    ;A PARITY ABORT DOESN'T OCCUR
1839  006746  005737  001642                  TST     @#CPU40           ;ARE WE ON AN 11/40?
1840  006752  001402                          BEQ     3$                ;BRANCH IF NO
1841                                                                    ;SINCE PC WILL BE UPDATED ON THE
1842                                                                    ;PARITY ABORT
1843  006754  162700  000002                  SUB     #2,R0             ;DROP BACK 1 WORD ADDRESS
1844                                                                    ;SINCE THE PC WILL NOT BE
1845                                                                    ;UPDATED ON THE PARITY ABORT
1846  006760  010037  001332          3$:     MOV     R0,@#$GOODAT      ;STORE THE PC THAT SHOULD
1847                                                                    ;BE PUSHED ON THE STACK
1848                                                                    ;IF A PARITY ABORT OCCURS
1849  006764  004371  000000                  JSR     R3,@(R1)          ;GO TO PARITY MEMORY AREA
1850  006770  104001                          HLT     +1                ;DIDN'T ABORT
1851  006772  000406                          BR      .+16              ;GO TO RESET THE STACK
1852  006774  042777  000001  172620  F1:     BIC     #BIT0,@PARITY     ;DISABLE PARITY
1853  007002  004037  011550                  JSR     R0,@#CHECKLOC     ;CHECK FOR GOOD ABORT
1854  007006  104003                          HLT     +3                ;ABORTED INCORRECTLY
1855  007010  013706  001476                  MOV     @#NEWSTK,SP       ;RESET THE STACK
1856
1857
1858
                ;*********************************************************
                ;*********************************************************
                ;*********************************************************
                ;*********************************************************
                ;
                ;
                ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
                ;
                ;1ST PUSH = OLD R3 FROM THE 'JSR' (OTHER PARITY)
                ;
                ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
                ;       R3 CONTENTS (1ST PUSH)
                ;
                ;
                ;*********************************************************
                ;*********************************************************
                ;*********************************************************
                ;*********************************************************
1877
1878
```

```
1879                                    ;*****************************************************************
1880                                    ;TEST 43        TEST   RTS INSTRUCTION
1881                                    ;*****************************************************************
1882  007014  000004                   TST43:  SCOPE
1883                                    ;                       11/45 ***** ROM STATE 224 ****
1884                                    ;
1885                                    ;       11/40 **** ROM STATE 325 ****
1886  007016  004015                           JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1887  007020  007066                           G0                              ;ROUTINE ADDRESS
1888  007022  042777  000004  172572           BIC     #BIT2,@PARITY           ;WRITE NORMAL
1889  007030  012700  007056                   MOV     #G,R0                   ;SET UP A SUBROUTINE ADDRESS
1890  007034  012737  007052  001332           MOV     #.+16,@#$GDDAT          ;STORE THE PC THAT SHOULD
1891                                                                           ;BE PUSHED ON THE STACK
1892                                                                           ;IF A PARITY ABORT OCCURS
1893  007042  052777  000004  172552           BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
1894  007050  004320                           JSR     R3,(R0)+                ;SUBROUTINE CALL
1895  007052  104001                           HLT     +1                      ;DIDN'T ABORT
1896  007054  000412                           BR      .+26                    ;GO TO RESET THE STACK
1897  007056  042777  000004  172536  G:       BIC     #BIT2,@PARITY           ;WRITE NORMAL
1898  007064  000203                           RTS     R3                      ;RETURN FROM SUBROUTINE
1899  007066  042777  000001  172526  G0:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
1900  007074  004037  011550                   JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1901  007100  104003                           HLT     +3                      ;ABORTED INCORRECTLY
1902  007102  013706  001476                   MOV     @#NEWSTK,SP             ;RESET THE STACK
1903
1904
1905
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
        ;2ND PUSH = OLD PC FOR 'RTI' (OTHER PARITY)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
        ;       THE OLD PC (2ND PUSH)
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER
        ;       IS POSITIONED AT THE 1ST PUSH, THUS GIVING,
        ;       THE 'NEW' PS FOR THE RTI INSTRUCTION WHICH
        ;       WILL OVERLAY THE OLD PC SET UP FOR THE 'RTI'
        ;       INSTRUCTION (2ND PUSH) THUS GIVING,
        ;
        ;1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
        ;2ND PUSH = NEW PS FROM THE 'RTI'
        ;
        ;3RD PUSH = NEW PC FROM THE 'RTI'
        ;
        ;********************************************************************
        ;********************************************************************
```



```
        ;********************************************************************
        ;********************************************************************
1935
1936
1937                                    ;*****************************************************************
1938                                    ;TEST 44        TEST   1ST POP   RTI INSTRUCTION
1939                                    ;*****************************************************************
1940  007106  000004                   TST44:  SCOPE
1941                                    ;                       11/45 ***** ROM STATE 212 ****
1942                                    ;
1943                                    ;       11/40 **** ROM STATE 320 ****
1944  007110  004015                           JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1945  007112  007150                           H0                              ;ROUTINE ADDRESS
1946  007114  012746  000340                   MOV     #340,-(SP)              ;SET PS FOR 'RTI'
1947  007120  012746  007134                   MOV     #H1,-(SP)               ;SET RETURN FROM 'RTI'
1948  007124  012737  007134  001332           MOV     #H1,@#$GDDAT            ;STORE THE PC THAT SHOULD
1949                                                                           ;BE PUSHED ON THE STACK
1950                                                                           ;IF A PARITY ABORT OCCURS
1951  007132  000402                           BR      H                       ;GO DO 'RTI'
1952  007134  104001                   H1:     HLT     +1                      ;DIDN'T ABORT
1953  007136  000412                           BR      .+26                    ;GO TO RESET THE STACK
1954  007140  042777  000004  172454  H:       BIC     #BIT2,@PARITY           ;WRITE NORMAL
1955  007146  000002                           RTI                             ;RETURN FROM INTERRUPT
1956  007150  042777  000001  172444  H0:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
1957  007156  004037  011550                   JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1958  007162  104003                           HLT     +3                      ;ABORTED INCORRECTLY
1959  007164  013706  001476                   MOV     @#NEWSTK,SP             ;RESET THE STACK
1960
1961
1962
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
        ;2ND PUSH = OLD PC FOR 'RTI' (OTHER PARITY)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
        ;       THE OLD PS (1ST PUSH)
        ;
        ;       THE 2ND PUSH IS REWRITTEN NORMAL BEFORE
        ;       DOING THE 'RTI'
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK
        ;       POINTER IS PROPERLY UPDATED AND THE NEW
        ;       PS AND PC FROM THE RTI INSTRUCTION IS
        ;       PUSHED ONTO THE STACK, THUS GIVING,
        ;
        ;1ST PUSH = NEW PS FROM 'RTI' (NORMAL)
        ;2ND PUSH = NEW PC FROM 'RTI' (NORMAL)
```

```
                        ;
                        ;
                        ;********************************************************************
                        ;********************************************************************
                        ;********************************************************************
                        ;********************************************************************
 1993
 1994
 1995                                    ;*************************************************************
 1996                                    ;TEST 45        TEST   2ND POP   RTI INSTRUCTION
 1997                                    ;*************************************************************
 1998  007170  000004                    TST45:  SCOPE
 1999                                    ;                       11/45 **** ROM STATE 214 ****
 2000                                    ;
 2001                                    ;       11/40 **** ROM STATE 322 ****
 2002  007172  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2003  007174  007242                            H2                              ;ROUTINE ADDRESS
 2004  007176  012746  000340                    MOV     #340,-(SP)              ;SET PS FOR 'RTI'
 2005  007202  012746  007216                    MOV     #H3,-(SP)               ;SET RETURN FROM 'RTI'
 2006  007206  012737  007216  001332            MOV     #H3,@#$GDDAT            ;STORE THE PC THAT SHOULD
 2007                                                                            ;BE PUSHED ON THE STACK
 2008                                                                            ;IF A PARITY ABORT OCCURS
 2009  007214  000402                            BR      H4                      ;GO DO 'RTI'
 2010  007216  104001                    H3:     HLT     +1                      ;DIDN'T ABORT
 2011  007220  000416                            BR      .+36                    ;GO TO RESET THE STACK
 2012  007222  042777  000005  172372    H4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE PARITY
 2013  007230  012646                            MOV     (SP)+,-(SP)             ;WRITE 1ST "POP" NORMAL SO NO
 2014                                                                            ;ERROR WILL OCCUR
 2015  007232  052777  000001  172362            BIS     #BIT0,@PARITY           ;ENABLE PARITY
 2016  007240  000002                            RTI                             ;RETURN FROM INTERRUPT
 2017  007242  042777  000001  172352    H2:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
 2018  007250  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2019  007254  104003                            HLT     +3                      ;ABORTED INCORRECTLY
 2020  007256  012706  001100                    MOV     #STACK,SP               ;INITIALIZE THE STACK BACK
 2021                                                                            ;TO 1100 FOR WE DON'T NEED IT
 2022                                                                            ;TO BE IN PARITY MEMORY AREA
 2023                                                                            ;ANYMORE
 2024                                    ;*************************************************************
 2025                                    ;TEST 46        TEST (DATA)   SM6,DM0   ADD INSTRUCTION
 2026                                    ;*************************************************************
 2027  007262  000004                    TST46:  SCOPE
 2028                                    ;                       11/45 **** ROM STATE 142 ****
 2029                                    ;
 2030                                    ;       11/40 **** ROM STATE 250 ****
 2031  007264  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2032  007266  007320                            L                               ;ROUTINE ADDRESS
 2033  007270  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
 2034  007272  010020                            MOV     R0,(R0)+                ;DO THE DATO
 2035  007274  012737  007306  001332            MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2036                                                                            ;BE PUSHED ON THE STACK
 2037                                                                            ;IF A PARITY ABORT OCCURS
 2038  007302  066000  177776                    ADD     -2(R0),R0               ;DO A DATI
 2039  007306  042777  000004  172306            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
 2040  007314  104001                            HLT     +1                      ;DIDN'T ABORT
```



```
 2041  007316  000410                            BR      .+22                    ;GO TO NEXT TEST
 2042  007320  042777  000005  172274    L:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
 2043  007326  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2044  007332  104003                            HLT     +3                      ;ABORTED INCORRECTLY
 2045  007334  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
 2046                                    ;*************************************************************
 2047                                    ;TEST 47        TEST  DM1  TSTB INSTRUCTION
 2048                                    ;*************************************************************
 2049  007340  000004                    TST47:  SCOPE
 2050                                    ;                       11/45 **** ROM STATE 175 ****
 2051                                    ;
 2052                                    ;       11/40 **** ROM STATE 267 ****
 2053  007342  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2054  007344  007374                            M                               ;ROUTINE ADDRESS
 2055  007346  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
 2056  007350  010010                            MOV     R0,(R0)                 ;DO THE DATO
 2057  007352  012737  007362  001332            MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2058                                                                            ;BE PUSHED ON THE STACK
 2059                                                                            ;IF A PARITY ABORT OCCURS
 2060  007360  105710                            TSTB    (R0)                    ;DO A DATI, DATIP
 2061  007362  042777  000004  172232            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
 2062  007370  104001                            HLT     +1                      ;DIDN'T ABORT
 2063  007372  000410                            BR      .+22                    ;GO TO NEXT TEST
 2064  007374  042777  000005  172220    M:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
 2065  007402  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2066  007406  104003                            HLT     +3                      ;ABORTED INCORRECTLY
 2067  007410  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
 2068                                    ;*************************************************************
 2069                                    ;TEST 50        TEST  DM3  CLRB INSTRUCTION
 2070                                    ;*************************************************************
 2071  007414  000004                    TST50:  SCOPE
 2072                                    ;                       11/45 **** ROM STATE 221 ****
 2073                                    ;
 2074                                    ;       11/40 **** ROM STATE 264 ****
 2075  007416  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2076  007420  007450                            NN                              ;ROUTINE ADDRESS
 2077  007422  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
 2078  007424  010010                            MOV     R0,(R0)                 ;DO THE DATO
 2079  007426  012737  007436  001332            MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2080                                                                            ;BE PUSHED ON THE STACK
 2081                                                                            ;IF A PARITY ABORT OCCURS
 2082  007434  105030                            CLRB    @(R0)+                  ;DO A DATI
 2083  007436  042777  000004  172156            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
 2084  007444  104001                            HLT     +1                      ;DIDN'T ABORT
 2085  007446  000410                            BR      .+22                    ;GO TO NEXT TEST
 2086  007450  042777  000005  172144    NN:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
 2087  007456  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2088  007462  104003                            HLT     +3                      ;ABORTED INCORRECTLY
 2089  007464  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
 2090                                    ;*************************************************************
 2091                                    ;TEST 51        TEST  SM1  SUBTRACT INSTRUCTION
 2092                                    ;*************************************************************
 2093  007470  000004                    TST51:  SCOPE
 2094                                    ;                       11/45 **** ROM STATE 27 ****
```

```
2095                                    ;
2096                                    ;       11/40 **** ROM STATE 250 ****
2097  007472  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2098  007474  007526                            P                               ;ROUTINE ADDRESS
2099  007476  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
2100  007500  010010                            MOV     R0,(R0)                 ;DO THE DATO
2101  007502  012737  007512  001332            MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
2102                                                                            ;BE PUSHED ON THE STACK
2103                                                                            ;IF A PARITY ABORT OCCURS
2104  007510  161027  177777                   SUB     (R0),#177777            ;DO A DATI
2105  007514  042777  000004  172100            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2106  007522  104001                            HLT     +1                      ;DIDN'T ABORT
2107  007524  000410                            BR      .+22                    ;GO TO NEXT TEST
2108  007526  042777  000005  172066  P:        BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2109  007534  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2110  007540  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2111  007542  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2112                                    ;*****************************************************************
2113                                    ;TEST 52        TEST (DATA)  SM5  BISB INSTRUCTION
2114                                    ;*****************************************************************
2115  007546  000004                    TST52:  SCOPE
2116                                    ;                 11/45 **** ROM STATE 177 ****
2117                                    ;
2118                                    ;       11/40 **** ROM STATE 245 ****
2119  007550  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2120  007552  007604                            R                               ;ROUTINE ADDRESS
2121  007554  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
2122  007556  010010                            MOV     R0,(R0)                 ;DO THE DATO
2123  007560  012737  007570  001332            MOV     #.+10,@#$$DDAT          ;STORE THE PC THAT SHOULD
2124                                                                            ;BE PUSHED ON THE STACK
2125                                                                            ;IF A PARITY ABORT OCCURS
2126  007566  135060  000002                   BISB    @-(R0),2(R0)            ;DO A DATI
2127  007572  042777  000004  172022            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2128  007600  104001                            HLT     +1                      ;DIDN'T ABORT
2129  007602  000410                            BR      .+22                    ;GO TO NEXT TEST
2130  007604  042777  000005  172010  R:        BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2131  007612  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2132  007616  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2133  007620  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2134                                    ;*****************************************************************
2135                                    ;TEST 53        NEXT 4 TESTS ARE NON-INTERLEAVE 4K DEPENDENT
2136                                    ;*****************************************************************
2137  007624  000004                    TST53:  SCOPE
2138  007626  022701  017360                    CMP     #17360,R1               ;HAVE WE PARITY IN THE LOWER 4K?
2139                                                                            ;AND IS THE SELECTED REGISTER
2140                                                                            ;GOVERNING THE 4K AREA?
2141                                                                            ;THIS COMPARE IS IF THE KT11
2142                                                                            ;ISN'T ENABLED DURING PROGRAM
2143                                                                            ;EXECUTION
2144  007632  101005                            BHI     2$                      ;YES - PROCEED TO NEXT TESTS
2145                                                                            ;WHICH DEPEND ON PARITY IN LOWER 4K
2146  007634  022777  000140  171554            CMP     #140,@$SEIAD            ;THE ABOVE COMPARE DIDN'T
2147                                                                            ;CHECK - - DO THIS COMPARE TO
2148                                                                            ;SEE IF IT WAS BECAUSE THE
```



```
2149                                                                            ;TABLE WAS CREATED WITH MEMORY
2150                                                                            ;MANAGEMENT TURNED ON
2151  007642  001401                            BEQ     2$                      ;PROCEED TO NEXT TESTS IF THIS
2152                                                                            ;COMPARE CHECKS
2153  007644  000404                            BR      4$                      ;THE REGISTER UNDER TEST IS NOT
2154                                                                            ;CONTROLLING THE LOWER 4K!!!
2155                                                                            ;GO TO RE-EVALUATE $TSTNM AND
2156                                                                            ;JUMP OVER THE (4) 4K DEPENDENT
2157                                                                            ;TESTS
2158  007646  022777  000001  171572  2$:       CMP     #1,@INTERAD             ;IS THIS CONTROLLER
2159                                                                            ;INTERLEAVED??
2160  007654  001405                            BEQ     3$                      ;BRANCH IF NO
2161  007656  062767  000004  171416  4$:       ADD     #4,$TSTNM               ;SET $TSTNM TO PROPER VALUE
2162                                                                            ;SINCE THE NEXT 4 TESTS WILL
2163                                                                            ;BE SKIPPED
2164  007664  000137  010426                    JMP     @#RED+26                ;JUMP TO 1ST INDEX WORD TEST
2165  007670  005237  001624            3$:     INC     @#PSPCORZONES           ;SET FLAG INDICATING TO 'CHECKLOC'
2166                                                                            ;ROUTINE THAT PS AND PC FETCH
2167                                                                            ;AND RED & YELLOW ZONE AREAS
2168                                                                            ;ARE GOING TO BE TESTED
2169
2170
2171
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - OLD PC FROM ERROR TRAP (NORMAL)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PS
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
        ;       ALTERED FROM THE ORIGINAL ERROR TRAP, THUS GIVING,
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - NEW PS FROM THE PARITY ERROR
        ;
        ;3RD PUSH - NEW PC FROM THE PARITY ERROR
        ;
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
        ;********************************************************************
2197
2198
2199                                    ;*****************************************************************
2200                                    ;TEST 54        TEST NEW PS FETCH
2201                                    ;*****************************************************************
2202  007674  000004                    TST54:  SCOPE
```

```
2203                                    ;               11/45 **** ROM STATE 357 ****
2204                                    ;
2205                                    ;       11/40 **** ROM STATE 115 ****
2206  007676  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2207  007700  007754                            T0                              ;ROUTINE ADDRESS
2208  007702  042777  000004  171712            BIC     #BIT2,@PAHCR            ;WRITE NORMAL
2209  007710  012737  007750  000010            MOV     #T,@#RESVEC             ;RESERVED INSTRUCTION TIMEOUT
2210                                                                            ;VECTOR ADDRESS
2211  007716  012737  007750  001332            MOV     #T,@#$GOOAT             ;STORE THE PC THAT SHOULD
2212                                                                            ;BE PUSHED ON THE STACK
2213                                                                            ;IF A PARITY ABORT OCCURS
2214  007724  052777  000004  171670            BIS     #BIT2,@PAHCR            ;WRITE OTHER PARITY
2215  007732  012737  000340  000012            MOV     #340,@#RESVEC+2         ;NEW 'PS' FOR ERROR TRAP
2216  007740  042777  000004  171654            BIC     #BIT2,@PAHCR            ;WRITE NORMAL
2217  007746  007000                            7000                            ;NON-RECOGNIZABLE OP-CODE
2218                                                                            ;SHOULD ATTEMPT TO TRAP TO 'T:'
2219  007750  104001                    T:      HLT     +1                      ;DIDN'T ABORT
2220  007752  000406                            BR      .+16                    ;GO TO RESET THE STACK
2221  007754  042777  000001  171640    T0:     BIC     #BIT0,@PAHCR            ;DISABLE PARITY
2222  007762  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2223  007766  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2224  007770  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2225
2226
2227
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - OLD PC FROM ERROR TRAP (NORMAL)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PC
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
        ;       NOT ALTERED FROM THE ORIGINAL ERROR TRAP AND THE
        ;       PC FOR THE PARITY ERROR IS THE OLD PS, THUS GIVING,
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;
        ;IN OTHER WORDS THE ORIGINAL ERROR TRAP AND VECTOR IS LOST!!!!
        ;
        ;
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
2255
2256
```



```
2257                                    ;*********************************************************
2258                                    ;TEST 55        TEST NEW 'PC' FETCH
2259                                    ;*********************************************************
2260  007774  000004                    TST55:  SCOPE
2261                                    ;               11/45 **** ROM STATE 355 ****
2262                                    ;
2263                                    ;       11/40 **** ROM STATE 333 ****
2264  007776  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2265  010000  010056                            V0                              ;ROUTINE ADDRESS
2266  010002  012737  010052  000010            MOV     #V,@#RESVEC             ;RESERVED INSTRUCTION TIMEOUT
2267                                                                            ;VECTOR ADDRESS
2268  010010  012737  000340  000012            MOV     #340,@#RESVEC+2         ;NEW 'PS' FOR ERROR TRAP
2269  010016  005737  001642                    TST     @#CPU40                 ;ARE WE ON AN 11/40?
2270  010022  001404                            BEQ     1$                      ;BRANCH IF NO
2271  010024  012737  010052  001332            MOV     #V,@#$GOOAT             ;STORE THIS VALUE OF THE PC THAT
2272                                                                            ;SHOULD BE PUSHED ON THE STACK
2273                                                                            ;IF A PARITY ABORT OCCURS
2274                                                                            ;SINCE THE PC IS UPDATED FIRST
2275                                                                            ;THEN THE VECTOR DATA TAKEN
2276  010032  000403                            BR      2$                      ;CONTINUE WITH TEST
2277  010034  012737  000010  001332    1$:     MOV     #10,@#$GOOAT            ;STORE THE PC THAT SHOULD
2278                                                                            ;BE PUSHED ON THE STACK
2279                                                                            ;IF A PARITY ABORT OCCURS
2280  010042  042777  000004  171552    2$:     BIC     #BIT2,@PAHCR            ;WRITE NORMAL
2281  010050  007000                            7000                            ;NON-RECOGNIZABLE OP-CODE
2282                                                                            ;SHOULD ATTEMPT TO TRAP TO 'V:'
2283  010052  104001                    V:      HLT     +1                      ;DIDN'T ABORT
2284  010054  000406                            BR      .+16                    ;GO TO RESET THE STACK
2285  010056  042777  000001  171536    V0:     BIC     #BIT0,@PAHCR            ;DISABLE PARITY
2286  010064  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2287  010070  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2288  010072  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2289
2290
2291
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;
        ;
        ;THE FOLLOWING TEST WILL/SHOULD CAUSE A PARITY ABORT IN THE
        ;'YELLOW' ZONE.  THE 'YELLOW' ZONE IS AN AREA UP TO A 16 WORD
        ;LOCATION LIMIT BEYOND THE STACK OVERFLOW BOUNDARY OF 400
        ;(OCTAL).  I.E. LOCATIONS 376 - 340 COMPRISE THE YELLOW ZONE.
        ;
        ;SINCE PARITY ERRORS HAVE HIGHEST PRIORITY WE WILL BE LOOKING
        ;FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
        ;TRAP TO LOCATION 4.
        ;
        ;
        ;THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
        ;SHOULD BE AS FOLLOWS:
        ;
```

```
        ;1ST PUSH - PS FROM THE PARITY ERROR
        ;2ND PUSH - PC FROM THE PARITY ERROR
        ;3RD PUSH - PS FROM THE STACK VIOLATION
        ;4TH PUSH - PC FROM THE STACK VIOLATION
        ;
        ;NOTE:  THE ABOVE CONTENTS WILL EXIST ON THE STACK IF BOTH
        ;       THE PARITY ERROR AND THE STACK VIOLATION WERE
        ;       RECOGNIZED AND THE PARITY ERROR TOOK PRECEDENCE!
        ;
        ;
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
  2325
  2326
  2327                                  ;*********************************************************
  2328                                  ;TEST 56        TEST ABORT IN 'YELLOW' ZONE
  2329                                  ;*********************************************************
  2330  010076  000004                 TST56:  SCOPE
  2331                                  ;                 11/45 **** ROM STATE 177 ****
  2332                                  ;
  2333                                  ;       11/40 **** ROM STATE 267 ****
  2334  010100  004015                         JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
  2335  010102  010162                         V1                              ;ROUTINE ADDRESS
  2336  010104  005037  000340                 CLR     @#340                   ;CLEAR BOTTOM LIMIT LOCATION
  2337                                                                          ;OF 'YELLOW' ZONE USING 'OTHER'
  2338                                                                          ;PARITY
  2339  010110  042777  000004  171504         BIC     #BIT2,@PARITY           ;WRITE NORMAL
  2340  010116  012706  000376                 MOV     #376,SP                 ;SET STACK IN 'YELLOW' AREA
  2341  010122  013727  000004                 MOV     @#4,(PC)+               ;SAVE CONTENTS OF LOC. 4 IN
  2342                                                                          ;NEXT LOCATION
  2343  010126  000000                 SAVLOC4:        .WORD   0               ;ORIGINAL CONTENTS OF LOC. 4
  2344                                                                          ;GO HERE
  2345  010130  012737  010166  000004         MOV     #V2,@#ERRVEC            ;SET UP A TIMEOUT VECTOR SERVICE
  2346                                                                          ;ROUTINE ADDRESS FOR STACK VIO-
  2347                                                                          ;LATION
  2348  010136  012737  000340  000006         MOV     #340,@#ERRVEC+2         ;NEW PS ON TIMEOUT TRAP
  2349  010144  012737  010156  001332         MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
  2350                                                                          ;BE PUSHED ON THE STACK
  2351                                                                          ;IF A PARITY ABORT OCCURS
  2352  010152  005766  177742                 TST     -36(SP)                 ;REFERENCE BOTTOM LIMIT OF
  2353                                                                          ;'YELLOW' ZONE USING REGISTER 6
  2354                                                                          ;THIS INSTRUCTION SHOULD CAUSE
  2355                                                                          ;AN ABORT 1ST, THEN A STACK
  2356                                                                          ;VIOLATION
  2357  010156  104010                         HLT     +10                     ;DIDN'T ABORT OR RECOGNIZE THE
  2358                                                                          ;STACK VIOLATION
  2359  010160  000421                         BR      YELLOW                  ;GO TO RESET STACK AND RESTORE
  2360                                                                          ;TIMEOUT VECTORS
  2361  010162  104011                 V1:     HLT     +11                     ;ABORTED BUT STACK VIOLATION
  2362                                                                          ;NOT RECOGNIZED
  2363  010164  000417                         BR      YELLOW                  ;GO TO RESET STACK AND RESTORE
  2364                                                                          ;TIMEOUT VECTORS
```



```
  2365  010166  042777  000001  171426 V2:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
  2366  010174  005737  000372               TST     @#372                   ;STACK VIOLATION PICKED UP -
  2367                                                                          ;WAS THE PARITY ABORT?
  2368  010200  001002                         BNE     1$                      ;BRANCH IF YES
  2369  010202  104012                         HLT     +12                     ;STACK VIOLATION PICKED UP
  2370                                                                          ;BUT ABORT NOT RECOGNIZED
  2371  010204  000407                         BR      YELLOW                  ;GO TO RESET STACK AND RESTORE
  2372                                                                          ;TIMEOUT VECTORS
  2373  010206  012706  001100         1$:     MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
  2374  010212  013746  000372                 MOV     @#372,-(SP)             ;PUSH ABORT PC ONTO KERNAL
  2375                                                                          ;STACK OUT OF VIOLATION AREAS
  2376                                                                          ;FOR CHECKING PURPOSES
  2377  010216  004037  011550                 JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
  2378  010222  104003                         HLT     +3                      ;ABORTED INCORRECTLY
  2379  010224  012706  001100         YELLOW: MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
  2380  010230  013737  010126  000004         MOV     @#SAVLOC4,@#4           ;RESTORE CONTENTS OF LOC. 4
  2381  010236  005037  000006                 CLR     @#6                     ;RESTORE CONTENTS OF LOC. 6
  2382  010242  005037  000372                 CLR     @#372                   ;RESET TRAPCATCHER LOCATION
  2383                                                                          ;FOR NEXT TEST
  2384  010246  005037  000340                 CLR     @#340                   ;CLEAR BOTTOM LIMIT LOCATION
  2385                                                                          ;OF 'YELLOW' ZONE USING NORMAL
  2386                                                                          ;PARITY
  2387
  2388
  2389
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;
        ;
        ;THE FOLLOWING TEST WILL/SHOULD CAUSE A PARITY ABORT IN THE
        ;'RED' ZONE.  THE 'RED' ZONE IS THE AREA BEYOND THE 'YELLOW'
        ;ZONE DESCRIBED IN THE ABOVE TEST.  I.E. LOCATIONS 336 ON
        ;DOWN COMPRISE THE 'RED' ZONE
        ;
        ;SINCE PARITY ERRORS HAVE HIGHEST PRIORITY WE WILL BE LOOKING
        ;FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
        ;TRAP TO LOCATION 4.
        ;
        ;
        ;THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
        ;SHOULD BE AS FOLLOWS:
        ;
        ;LOC. 0 - PC FROM STACK VIOLATION
        ;LOC. 2 - PS FROM STACK VIOLATION
        ;
        ;NOTE:  THE PS AND PC FROM THE STACK VIOLATION ARE IN LOCS. 0 &2
        ;       BECAUSE THE STACK POINTER (R6) IS REPOSITIONED TO LOC. 4
        ;       (BY HARDWARE)!!
        ;
```

```
                ;LOC. 374 = PS FROM PARITY ABORT
                ;LOC. 372 = PC FROM PARITY ABORT
                ;
                ;NOTE:  THE ABOVE CONTENTS WILL EXIST IN THE 2 DIFFERENT STACK
                ;       AREAS IF BOTH THE PARITY ERROR AND STACK VIOLATION WERE
                ;       RECOGNIZED.  THE TEST BELOW WILL DIFFERENTIATE BETWEEN
                ;       WHICH OCCURRED FIRST.
                ;
                ;
                ;*********************************************************************
                ;*********************************************************************
                ;*********************************************************************
                ;*********************************************************************
 2429
 2430
 2431                                   ;*********************************************************************
 2432                                   ;TEST 57        TEST ABORT IN 'RED' ZONE
 2433                                   ;*********************************************************************
 2434  010252  000004                  TST57:  SCOPE
 2435                                   ;                       11/45 ***** ROM STATE 177 ****
 2436                                   ;
 2437                                   ;       11/40 **** ROM STATE 267 ****
 2438  010254  004015                          JSR     R0,(R5)                 ;SET UP A PARITY VECTOR SERVICE
 2439  010256  010336                          V3                              ;ROUTINE ADDRESS
 2440  010260  005037  000336                  CLR     @#336                   ;CLEAR 1ST LOCATION OF 'RED'
 2441                                                                           ;ZONE USING 'OTHER' PARITY
 2442  010264  042777  000004  171330          BIC     #BIT2,@PAMITY           ;WRITE NORMAL
 2443  010272  012706  000376                  MOV     #376,SP                 ;SET STACK IN 'YELLOW'AREA
 2444  010276  013727  000004                  MOV     @#4,(PC)+               ;SAVE CONTENTS OF LOC. 4
 2445                                                                           ;IN NEXT LOCATION
 2446  010302  000000                  HOLDLOC4:       .WORD   0               ;ORIGINAL CONTENTS OF LOC. 4
 2447                                                                           ;GO HERE
 2448  010304  012737  010342  000004          MOV     #V4,@#ERRVEC            ;SET UP A TIMEOUT VECTOR SERVICE
 2449                                                                           ;ROUTINE ADDRESS FOR STACK
 2450                                                                           ;VIOLATION
 2451  010312  012737  000340  000006          MOV     #340,@#ERRVEC+2         ;NEW PS ON TIMEOUT TRAP
 2452  010320  012737  010332  001332          MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2453                                                                           ;BE PUSHED ON THE STACK
 2454                                                                           ;IF A PARITY ABORT OCCURS
 2455  010326  005766  177740                  TST     -40(SP)                 ;REFERENCE 1ST LOCATION OF 'RED'
 2456                                                                           ;ZONE USING REGISTER 6
 2457                                                                           ;THIS INSTRUCTION SHOULD CAUSE
 2458                                                                           ;AN ABORT 1ST, THEN A STACK
 2459                                                                           ;VIOLATION
 2460  010332  104010                          HLT     +10                     ;DIDN'T ABORT OR RECOGNIZE THE
 2461                                                                           ;STACK VIOLATION
 2462  010334  000421                          BR      RED                     ;GO TO RESET STACK AND RESTORE
 2463                                                                           ;TIMEOUT VECTORS
 2464  010336  104011                  V3:     HLT     +11                     ;ABORTED BUT STACK VIOLATION
 2465                                                                           ;NOT RECOGNIZED
 2466  010340  000417                          BR      RED                     ;GO TO RESET STACK AND RESTORE
 2467                                                                           ;TIMEOUT VECTORS
 2468  010342  042777  000001  171252  V4:     BIC     #BIT0,@PAMITY           ;DISABLE PARITY
 2469  010350  005737  000372                  TST     @#372                   ;STACK VIOLATION PICKED UP =
```



```
 2470                                                                           ;WAS THE PARITY ABORT?
 2471  010354  001002                          BNE     1$                      ;BRANCH IF YES
 2472  010356  104012                          HLT     +12                     ;STACK VIOLATION PICKED UP BUT
 2473                                                                           ;ABORT NOT RECOGNIZED
 2474  010360  000407                          BR      RED                     ;GO TO RESET STACK AND RESTORE
 2475                                                                           ;TIMEOUT VECTORS
 2476  010362  012706  001100          1$:     MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
 2477  010366  013746  000372                  MOV     @#372,-(SP)             ;PUSH ABORT PC ONTO KERNAL
 2478                                                                           ;STACK OUT OF VIOLATION AREAS
 2479                                                                           ;FOR CHECKING PURPOSES
 2480  010372  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2481  010376  104003                          HLT     +3                      ;ABORTED INCORRECTLY
 2482  010400  012706  001100          RED:    MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
 2483  010404  013737  010302  000004          MOV     @#HOLDLOC4,@#4          ;RESTORE CONTENTS OF LOC. 4
 2484  010412  005037  000006                  CLR     @#6                     ;RESTORE CONTENTS OF LOC. 6
 2485  010416  005037  000372                  CLR     @#372                   ;RESET TRAPCATCHER LOCATION
 2486                                                                           ;FOR NEXT TEST
 2487  010422  005037  000336                  CLR     @#336                   ;CLEAR 1ST LOCATION OF 'RED'
 2488                                                                           ;ZONE USING NORMAL PARITY
 2489                                   ;*********************************************************************
 2490                                   ;TEST 60        TEST (INDEX WORD)  SM0,DM7  MOV INSTRUCTION
 2491                                   ;*********************************************************************
 2492  010426  000004                  TST60:  SCOPE
 2493                                   ;                       11/45 ***** ROM STATE 7 ****
 2494                                   ;
 2495                                   ;       11/40 **** ROM STATE 206 ****
 2496  010430  005037  001624                  CLR     @#PSPCORZONES           ;CLEAR PS, PC OR ZONES
 2497                                                                           ;ABORT FLAG
 2498  010434  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2499  010436  010514                          W                               ;ROUTINE ADDRESS
 2500  010440  042777  000004  171154          BIC     #BIT2,@PAMITY           ;WRITE NORMAL
 2501  010446  010100                          MOV     R1,R0                   ;SET UP FOR A DATO
 2502  010450  012720  010070                  MOV     #10070,(R0)+            ;MOVE THE INSTRUCTION
 2503                                                                           ;'MOV R0,@-2(R0)' TO PARITY
 2504                                                                           ;MEMORY AREA
 2505  010454  052777  000004  171140          BIS     #BIT2,@PAMITY           ;WRITE OTHER PARITY
 2506  010462  012720  177776                  MOV     #-2,(R0)+               ;WRITE THE INDEX WORD IN NEXT
 2507                                                                           ;PARITY MEMORY AREA LOCATION
 2508  010466  042777  000004  171126          BIC     #BIT2,@PAMITY           ;WRITE NORMAL
 2509  010474  010037  001332                  MOV     R0,@#$GDDAT             ;STORE THE PC THAT SHOULD
 2510                                                                           ;BE PUSHED ON THE STACK
 2511                                                                           ;IF A PARITY ABORT OCCURS
 2512  010500  012710  000203                  MOV     #203,(R0)               ;MOVE 'RTS R3' INTO NEXT
 2513                                                                           ;PARITY MEMORY AREA LOCATION
 2514                                                                           ;IN CASE WE DON'T ABORT
 2515  010504  004360  177774                  JSR     R3,-4(R0)               ;GO TO PARITY MEMORY AREA
 2516  010510  104001                          HLT     +1                      ;DIDN'T ABORT
 2517  010512  000410                          BR      .+22                    ;GO TO NEXT TEST
 2518  010514  042777  000001  171100  W:      BIC     #BIT0,@PAMITY           ;DISABLE PARITY
 2519  010522  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2520  010526  104003                          HLT     +3                      ;ABORTED INCORRECTLY
 2521  010530  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
 2522                                   ;*********************************************************************
 2523                                   ;TEST 61        TEST (INDEX WORD)  SM6,DM0  CMP INSTRUCTION
```

```
2524                                    ;*************************************************************
2525  010534  000004                    TST61:  SCOPE
2526                                    ;                 11/45 ***** ROM STATE 26 ****
2527                                    ;
2528                                    ;       11/40 **** ROM STATE 241 ****
2529  010536  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2530  010540  010616                            XX                              ;ROUTINE ADDRESS
2531  010542  042777  000004  171052            BIC     #BIT2,@PARITY           ;WRITE NORMAL
2532  010550  010100                            MOV     R1,R0                   ;SET UP FOR A DATO
2533  010552  012720  026000                    MOV     #26000,(R0)+            ;MOVE THE INSTRUCTION
2534                                                                            ;'CMP -2(R0),R0' TO PARITY
2535                                                                            ;MEMORY AREA
2536  010556  052777  000004  171036            BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
2537  010564  012720  177776                    MOV     #-2,(R0)+               ;WRITE THE INDEX WORD IN NEXT
2538                                                                            ;PARITY MEMORY AREA LOCATION
2539  010570  042777  000004  171024            BIC     #BIT2,@PARITY           ;WRITE NORMAL
2540  010576  010037  001332                    MOV     R0,@#$GDDAT             ;STORE THE PC THAT SHOULD
2541                                                                            ;BE PUSHED ON THE STACK
2542                                                                            ;IF A PARITY ABORT OCCURS
2543  010602  012710  000203                    MOV     #203,(R0)               ;MOVE 'RTS R3' INTO NEXT
2544                                                                            ;PARITY MEMORY AREA LOCATION
2545                                                                            ;IN CASE WE DON'T ABORT
2546  010606  004360  177774                    JSR     R3,-4(R0)               ;GO TO PARITY MEMORY AREA
2547  010612  104001                            HLT     +1                      ;DIDN'T ABORT
2548  010614  000410                            BR      .+22                    ;GO TO NEXT TEST
2549  010616  042777  000001  170776  XX:       BIC     #BIT0,@PARITY           ;DISABLE PARITY
2550  010624  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2551  010630  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2552  010632  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2553                                    ;*************************************************************
2554                                    ;TEST 62        TEST (INDEX WORD)  SM0,DM6  MOV INSTRUCTION
2555                                    ;*************************************************************
2556  010636  000004                    TST62:  SCOPE
2557                                    ;                 11/45 ***** ROM STATE 6 ****
2558                                    ;
2559                                    ;       11/40 **** ROM STATE 206 ****
2560  010640  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2561  010642  010720                            Y                               ;ROUTINE ADDRESS
2562  010644  042777  000004  170750            BIC     #BIT2,@PARITY           ;WRITE NORMAL
2563  010652  010100                            MOV     R1,R0                   ;SET UP FOR A DATO
2564  010654  012720  010060                    MOV     #10060,(R0)+            ;MOVE THE INSTRUCTION
2565                                                                            ;'MOV R0,-2(R0)' TO PARITY
2566                                                                            ;MEMORY AREA
2567  010660  052777  000004  170734            BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
2568  010666  012720  177776                    MOV     #-2,(R0)+               ;WRITE THE INDEX WORD IN NEXT
2569                                                                            ;PARITY MEMORY AREA LOCATION
2570  010672  042777  000004  170722            BIC     #BIT2,@PARITY           ;WRITE NORMAL
2571  010700  010037  001332                    MOV     R0,@#$GDDAT             ;STORE THE PC THAT SHOULD
2572                                                                            ;BE PUSHED ON THE STACK
2573                                                                            ;IF A PARITY ABORT OCCURS
2574  010704  012710  000203                    MOV     #203,(R0)               ;MOVE 'RTS R3' INTO NEXT
2575                                                                            ;PARITY MEMORY AREA LOCATION
2576                                                                            ;IN CASE WE DON'T ABORT
2577  010710  004360  177774                    JSR     R3,-4(R0)               ;GO TO PARITY MEMORY AREA
```



```
2578  010714  104001                            HLT     +1                      ;DIDN'T ABORT
2579  010716  000410                            BR      .+22                    ;GO TO NEXT TEST
2580  010720  042777  000001  170674  Y:        BIC     #BIT0,@PARITY           ;DISABLE PARITY
2581  010726  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2582  010732  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2583  010734  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2584                                    ;*************************************************************
2585                                    ;TEST 63        TEST   BPL INSTRUCTION (=BCOK)
2586                                    ;*************************************************************
2587  010740  000004                    TST63:  SCOPE
2588                                    ;                 11/45 ***** ROM STATE 321 ****
2589                                    ;
2590                                    ;       11/40 **** ROM STATE 1 ****
2591  010742  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2592  010744  011056                            Z0                              ;ROUTINE ADDRESS
2593  010746  042777  000004  170646            BIC     #BIT2,@PARITY           ;WRITE NORMAL
2594  010754  010102                            MOV     R1,R2                   ;SET UP FOR A DATO
2595  010756  162702  000004                    SUB     #4,R2                   ;CALCULATE THE START
2596                                                                            ;ADDRESS FOR THIS TEST
2597  010762  012722  012700                    MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
2598                                                                            ;'MOV #-1,R0' TO PARITY
2599                                                                            ;MEMORY AREA
2600  010766  012722  177777                    MOV     #-1,(R2)+               ;MOVE A -1 INTO NEXT
2601                                                                            ;PARITY MEMORY AREA LOCATION
2602  010772  012722  100001                    MOV     #100001,(R2)+           ;MOVE THE INSTRUCTION
2603                                                                            ;'BPL .+4' INTO NEXT PARITY
2604                                                                            ;MEMORY AREA LOCATION
2605  010776  052777  000004  170616            BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
2606  011004  005737  001642                    TST     @#CPU40                 ;ARE WE ON AN 11/40?
2607  011010  001405                            BEQ     2$                      ;BRANCH IF NO
2608                                                                            ;SINCE THE PC WILL BE UPDATED
2609                                                                            ;ON THE PARITY ABORT
2610  011012  010237  001332                    MOV     R2,@#$GDDAT             ;STORE THIS PC THAT SHOULD
2611                                                                            ;BE PUSHED ON THE STACK
2612                                                                            ;IF A PARITY ABORT OCCURS
2613                                                                            ;SINCE THE PC IS NOT UPDATED
2614                                                                            ;ON THE PARITY ABORT
2615  011016  012722  000240                    MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
2616                                                                            ;PARITY MEMORY AREA LOCATION
2617  011022  000404                            BR      1$                      ;CONTINUE WITH TEST
2618  011024  012722  000240            2$:     MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
2619                                                                            ;PARITY MEMORY AREA LOCATION
2620  011030  010237  001332                    MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
2621                                                                            ;BE PUSHED ON THE STACK
2622                                                                            ;IF A PARITY ABORT OCCURS
2623  011034  042777  000004  170560  1$:       BIC     #BIT2,@PARITY           ;WRITE NORMAL
2624  011042  012712  000203                    MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
2625                                                                            ;PARITY MEMORY AREA LOCATION
2626                                                                            ;IN CASE WE DON'T ABORT
2627  011046  004362  177770                    JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
2628  011052  104001                            HLT     +1                      ;DIDN'T ABORT
2629  011054  000410                            BR      .+22                    ;GO TO NEXT TEST
2630  011056  042777  000001  170536  Z0:       BIC     #BIT0,@PARITY           ;DISABLE PARITY
2631  011064  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
```

```
2632  011070  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2633  011072  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
2634                                  ;*************************************************************
2635                                  ;TEST 64        TEST   BNE INSTRUCTION   (=BCOK)
2636                                  ;*************************************************************
2637  011076  000004                  TST64:  SCOPE
2638                                  ;                       11/45 **** ROM STATE 322 ****
2639                                  ;
2640                                  ;       11/40 **** ROM STATE 1 ****
2641  011100  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2642  011102  011214                          Z1                              ;ROUTINE ADDRESS
2643  011104  042777  000004  170510          BIC     #BIT2,@PAKITY           ;WRITE NORMAL
2644  011112  010102                          MOV     R1,R2                   ;SET UP FOR A DATO
2645  011114  162702  000004                  SUB     #4,R2                   ;CALCULATE THE START ADDRESS
2646                                                                          ;FOR THIS TEST
2647  011120  012722  012700                  MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
2648                                                                          ;'MOV  #0,R0' INTO PARITY
2649                                                                          ;MEMORY AREA
2650  011124  012722  000000                  MOV     #0,(R2)+                ;MOVE A 0 INTO NEXT PARITY
2651                                                                          ;MEMORY AREA LOCATION
2652  011130  012722  001001                  MOV     #1001,(R2)+             ;MOVE THE INSTRUCTION
2653                                                                          ;'BNE .+4' INTO NEXT
2654                                                                          ;PARITY MEMORY AREA LOCATION
2655  011134  052777  000004  170460          BIS     #BIT2,@PAKITY           ;WRITE OTHER PARITY
2656  011142  005737  001642                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
2657  011146  001405                          BEQ     2$                      ;BRANCH IF NO
2658                                                                          ;SINCE THE PC WILL BE UPDATED
2659                                                                          ;ON THE PARITY ABORT
2660  011150  010237  001332                  MOV     R2,@#$GDDAT             ;STORE THIS PC THAT SHOULD
2661                                                                          ;BE PUSHED ON THE STACK
2662                                                                          ;IF A PARITY ABORT OCCURS
2663                                                                          ;SINCE THE PC WILL NOT BE
2664                                                                          ;UPDATED ON THE PARITY ABORT
2665  011154  012722  000240                  MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
2666                                                                          ;PARITY MEMORY AREA LOCATION
2667  011160  000404                          BR      1$                      ;CONTINUE WITH TEST
2668  011162  012722  000240          2$:     MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
2669                                                                          ;PARITY MEMORY AREA LOCATION
2670  011166  010237  001332                  MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
2671                                                                          ;BE PUSHED ON THE STACK
2672                                                                          ;IF A PARITY ABORT OCCURS
2673  011172  042777  000004  170422  1$:     BIC     #BIT2,@PAKITY           ;WRITE NORMAL
2674  011200  012712  000203                  MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
2675                                                                          ;PARITY MEMORY AREA LOCATION
2676                                                                          ;IN CASE WE DON'T ABORT
2677  011204  004362  177770                  JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
2678  011210  104001                          HLT     +1                      ;DIDN'T ABORT
2679  011212  000410                          BR      .+22                    ;GO TO NEXT TEST
2680  011214  042777  000001  170400  Z1:     BIC     #BIT0,@PAKITY           ;DISABLE PARITY
2681  011222  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2682  011226  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2683  011230  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
2684                                  ;*************************************************************
2685                                  ;TEST 65        TEST   BEW INSTRUCTION   (=BCOK)
```



```
2686                                  ;*************************************************************
2687  011234  000004                  TST65:  SCOPE
2688                                  ;                       11/45 **** ROM STATE 324 ****
2689                                  ;
2690                                  ;       11/40 **** ROM STATE 1 ****
2691  011236  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2692  011240  011352                          Z2                              ;ROUTINE ADDRESS
2693  011242  042777  000004  170352          BIC     #BIT2,@PAKITY           ;WRITE NORMAL
2694  011250  010102                          MOV     R1,R2                   ;SET UP FOR A DATO
2695  011252  162702  000004                  SUB     #4,R2                   ;CALCULATE THE START ADDRESS
2696                                                                          ;FOR THIS TEST
2697  011256  012722  012700                  MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
2698                                                                          ;'MOV #-1,R0' TO PARITY MEMORY
2699                                                                          ;AREA
2700  011262  012722  177777                  MOV     #-1,(R2)+               ;MOVE A -1 INTO NEXT PARITY
2701                                                                          ;MEMORY AREA LOCATION
2702  011266  012722  001401                  MOV     #1401,(R2)+             ;MOVE THE INSTRUCTION
2703                                                                          ;'BEQ .+4' INTO NEXT PARITY
2704                                                                          ;MEMORY AREA LOCATION
2705  011272  052777  000004  170322          BIS     #BIT2,@PAKITY           ;WRITE OTHER PARITY
2706  011300  005737  001642                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
2707  011304  001405                          BEQ     2$                      ;BRANCH IF NO
2708                                                                          ;SINCE THE PC WILL BE UPDATED
2709                                                                          ;ON THE PARITY ABORT
2710  011306  010237  001332                  MOV     R2,@#$GDDAT             ;STORE THIS PC THAT SHOULD
2711                                                                          ;BE PUSHED ON THE STACK
2712                                                                          ;IF A PARITY ABORT OCCURS
2713                                                                          ;SINCE THE PC WILL NOT BE
2714                                                                          ;UPDATED ON THE PARITY ABORT
2715  011312  012722  000240                  MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
2716                                                                          ;PARITY MEMORY AREA LOCATION
2717  011316  000404                          BR      1$                      ;CONTINUE WITH TEST
2718  011320  012722  000240          2$:     MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
2719                                                                          ;PARITY MEMORY AREA LOCATION
2720  011324  010237  001332                  MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
2721                                                                          ;BE PUSHED ON THE STACK
2722                                                                          ;IF A PARITY ABORT OCCURS
2723  011330  042777  000004  170264  1$:     BIC     #BIT2,@PAKITY           ;WRITE NORMAL
2724  011336  012712  000203                  MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
2725                                                                          ;PARITY MEMORY AREA LOCATION
2726                                                                          ;IN CASE WE DON'T ABORT
2727  011342  004362  177770                  JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
2728  011346  104001                          HLT     +1                      ;DIDN'T ABORT
2729  011350  000410                          BR      .+22                    ;GO TO NEXT TEST
2730  011352  042777  000001  170242  Z2:     BIC     #BIT0,@PAKITY           ;DISABLE PARITY
2731  011360  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2732  011364  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2733  011366  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
2734                                  ;*************************************************************
2735                                  ;TEST 66        END OF PRUGRAM
2736                                  ;*************************************************************
2737  011372  000004                  TST66:  SCOPE
2738  011374  042777  100005  170220          BIC     #BIT15!BIT2!BIT0,@#PARITY       ;DISABLE ALL PARITY
2739                                                                          ;AND CLEAR ERROR BIT!
```

```
2740  011402  005737  001630                  TST     @#USERTYPE              ;DID THE USER SELECT THE REGISTER?
2741  011406  001014                          BNE     1$                      ;BRANCH IF SO AND DON'T STEP
2742                                                                          ;UP THE TABLE
2743  011410  062737  000002  001336          ADD     #2,@#$REGAD             ;STEP UP TO NEXT REGISTER
2744  011416  062737  000002  001414          ADD     #2,@#$TMPAD             ;STEP UP TO CORRESPONDING
2745                                                                          ;PARITY MEMORY
2746  011424  062737  000002  001416          ADD     #2,@#$SETAD             ;STEP UP TO NEXT OFFSET - THIS
2747                                                                          ;IS ONLY APPLICABLE IF MEMORY
2748                                                                          ;MGMT IS TURNED ON
2749  011432  062737  000002  001446          ADD     #2,@#NTERAD             ;STEP UP TO NEXT INTERLEAVE
2750                                                                          ;VALUE
2751  011440  004337  011474          1$:     JSR     R3,@#FLAGSCLR           ;CLEAR PERTINENT FLAGS
2752  011444  000167  000320                  JMP     $EOP                    ;GO TO RING-A-DING
2753                                                                          ;BEFORE REITERATING THE PROGRAM
2754
2755
2756
2757                                  ;*************************************************************
2758                                  ;
2759                                  ;ROUTINE FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS
2760                                  ;
2761                                  ;*************************************************************
2762  011450  012077  170144          VECSET: MOV     (R0)+,@#INTVEC          ;WRITE ADDRESS INTO LOCATION 114
2763  011454  052777  000005  170140          BIS     #BIT2!BIT0,@#PARITY     ;WRITE OTHER PARITY AND DISABLE
2764  011462  000200                          RTS     R0                      ;RETURN TO TESTING
2765                                  ;*************************************************************
2766                                  ;
2767                                  ;ROUTINE TO RESET AND GO BACK TO TABLE BEGINNING
2768                                  ;
2769                                  ;*************************************************************
2770  011464  004337  011506          RESTART:        JSR     R3,@#INITIALIZE
2771  011470  000167  000274                  JMP     $EOP                    ;GO TO RING-A-DING
2772                                  ;*************************************************************
2773                                  ;
2774                                  ;ROUTINE TO CLEAR PERTINENT FLAGS BEFORE PASSING THRU THE PROGRAM
2775                                  ;WITH ANOTHER TABLE ENTRY
2776                                  ;
2777                                  ;*************************************************************
2778  011474  005037  001624          FLAGSCLR:       CLR     @#PSPCORZONES           ;CLEAR PS, PC AND ZONES ABORT
2779                                                                          ;FLAG
2780  011500  005037  001626                  CLR     @#MSREGFLAG             ;CLEAR MS11 REGISTER PRESENCE
2781                                                                          ;FLAG
2782  011504  000203                          RTS     R3                      ;RETURN
2783                                  ;*************************************************************
2784                                  ;
2785                                  ;ROUTINE TO COMPLETELY REINITIALIZE BEFORE RESTARTING PROGRAM OVER
2786                                  ;AT THE BEGINNING OF THE TABLE
2787                                  ;
2788                                  ;*************************************************************
2789  011506  005037  001624          INITIALIZE:     CLR     @#PSPCORZONES           ;CLEAR PS, PC AND ZONES
2790                                                                          ;ABORT FLAG
2791  011512  005037  001626                  CLR     @#MSREGFLAG             ;CLEAR MS11 REGISTER PRESENCE
2792                                                                          ;FLAG
2793  011516  012737  001340  001336          MOV     #$REG0,@#$REGAD         ;MOVE FIRST REGISTER CONTAINER
```



```
2794                                                                          ;INTO $REGAD WHICH IS USED TO
2795                                                                          ;POINT TO THE PARITY REGISTER
2796                                                                          ;TABLE
2797  011524  012737  001366  001414          MOV     #$TMP0,@#$TMPAD         ;MOVE FIRST MEMORY CONTAINER
2798                                                                          ;INTO $TMPAD WHICH IS USED TO
2799                                                                          ;POINT TO THE MEMORY LOCATION
2800                                                                          ;TABLE
2801  011532  012737  001420  001416          MOV     #$SET0,@#$SETAD         ;MOVE FIRST OFFSET CONTAINER
2802                                                                          ;INTO $SETAD WHICH IS USED TO
2803                                                                          ;POINT TO THE OFFSET LOCATION
2804                                                                          ;TABLE IF MEMORY MGMT IS USED
2805  011540  012737  001450  001446          MOV     #NTER0,@#NTERAD         ;MOVE 1ST INTERLEAVE CONTAINER
2806                                                                          ;INTO 'INTERAD' WHICH IS USED
2807                                                                          ;TO POINT TO THE INTERLEAVE
2808                                                                          ;LOCATION TABLE
2809  011546  000203                          RTS     R3                      ;RETURN
2810
2811
2812                                  ;*************************************************************
2813                                  ;
2814                                  ;SUBROUTINE TO CHECK THAT IF A TEST HAS ABORTED IT DID INDEED
2815                                  ;ABORT IN THE PROPER PLACE.  IT IS QUITE CONCEIVABLE THAT A TEST
2816                                  ;THAT SHOULD HAVE ABORTED IN THE TWO LOCATION MAP AREA ABORTED
2817                                  ;ON THE FETCH OF THE INSTRUCTION THAT WAS TO CAUSE THE ABORT.
2818                                  ;THIS SUBROUTINE WILL FLAG SUCH OCCURRENCES.  WITHOUT THIS CHECK
2819                                  ;THE PROGRAM WOULD APPEAR TO HAVE RUN PROPERLY.
2820                                  ;
2821                                  ;BOTH THE CORRECT HIGH ORDER ERROR ADDRESS BITS AND PROPER PC
2822                                  ;PUSH ON THE STACK ARE LOOKED FOR AFTER A PARITY ABORT OCCURS.
2823                                  ;
2824                                  ;*************************************************************
2825  011550  010246          CHECKLOC:       MOV     R2,-(SP)                ;SAVE R2 CONTENTS ON STACK
2826  011552  010346                  MOV     R3,-(SP)                ;SAVE R3 CONTENTS ON STACK
2827  011554  010446                  MOV     R4,-(SP)                ;SAVE R4 CONTENTS ON STACK
2828  011556  005002                  CLR     R2                      ;CLEAR ERROR ADDRESS BIT COMPARE
2829  011560  005004                  CLR     R4                      ;REGISTERS IN CASE WE HAVE AN OLD
2830                                                                          ;MOS DESIGN THAT DOESN'T HAVE
2831                                                                          ;ADDRESS BITS
2832  011562  005737  001626                  TST     @#MSREGFLAG             ;IS AN MF11 OR MS11 PARITY
2833                                                                          ;OPTION BEING TESTED?
2834  011566  001027                          BNE     7$                      ;BRANCH IF MS11 AND DON'T DO
2835                                                                          ;ADDRESS BITS CHECKING
2836  011570  005737  001624                  TST     @#PSPCORZONES           ;ARE WE DOING A PS OR PC FETCH
2837                                                                          ;OR ZONE ABORT TEST?
2838  011574  001402                          BEQ     5$                      ;BRANCH IF NO
2839  011576  005002                          CLR     R2                      ;SET THE PARITY REGISTER HIGH
2840                                                                          ;ORDER ADDRESS BITS VALUE
2841                                                                          ;(BITS 5 THRU 11) TO ZERO FOR
2842                                                                          ;THE PS OR PC FETCH ABORT TESTS
2843  011600  000414                          BR      2$                      ;GO CHECK IT AGAINST ACTUAL
2844                                                                          ;PARITY REGISTER VALUE
2845  011602  005737  002304          5$:     TST     @#$KT11                 ;MEMORY MANAGEMENT ON?
2846  011606  100003                          BPL     8$                      ;BRANCH IF NO
2847  011610  017702  167602                  MOV     @#$SETAD,R2             ;PICK UP THE OFFSET VALUE - IT
```

```
2848                                                                    ;SHOULD BE THE VALUE THAT WILL
2849                                                                    ;APPEAR IN THE PARITY REGISTER
2850                                                                    ;ERROR ADDRESS BITS
2851  011614  000406                          BR      2$                ;GO TO CHECK!
2852  011616  012703  017400          8$:     MOV     #17400,R3         ;GET A FIRST POSSIBLE ABORT
2853                                                                    ;LOCATION AREA
2854  011622  012702  000140                  MOV     #140,R2           ;GET THE FIRST POSSIBLE PARITY
2855                                                                    ;REGISTER HIGH ORDER ADDRESS
2856                                                                    ;BITS VALUE (BITS 5 THRU 11)
2857  011626  020103                  1$:     CMP     R1,R3             ;IS THIS THE ABORT AREA BEING
2858                                                                    ;USED?
2859  011630  101027                          BHI     3$                ;NO - SEE IF IT'S THE NEXT ONE
2860  011632  017704  167764          2$:     MOV     @PARITY,R4        ;PROCEED TO SEE IF
2861                                                                    ;IT WAS A PROPER ABORT BY LOOKING
2862                                                                    ;AT THE HIGH ORDER ADDRESS BITS
2863                                                                    ;OF THE PARITY REGISTER
2864                                                                    ;(BITS 5 THRU 11)
2865  011636  042704  170037                  BIC     #170037,R4        ;CLEAR ALL BITS EXCEPT 5 THRU 11
2866  011642  020402                          CMP     R4,R2             ;ARE THE ERROR ADDRESS BITS
2867                                                                    ;WHAT THEY SHOULD BE?
2868  011644  001026                          BNE     4$                ;BRANCH IF NO
2869  011646  010237  001326          7$:     MOV     R2,@#$GDAUR       ;STORE WHAT THE ADDRESS BITS
2870                                                                    ;SHOULD HAVE BEEN
2871  011652  010437  001330                  MOV     R4,@#$BDAUR       ;STORE WHAT THE ADDRESS BITS
2872                                                                    ;WERE
2873  011656  012604                          MOV     (SP)+,R4          ;RESTORE R4 CONTENTS
2874  011660  012603                          MOV     (SP)+,R3          ;RESTORE R3 CONTENTS
2875  011662  012602                          MOV     (SP)+,R2          ;RESTORE R2 CONTENTS
2876
2877                                  ;IF WE HAVE REACHED THIS PATH 1 OF 2 CONDITIONS EXIST --
2878                                  ;WE HAVE AN OLD MOS DESIGN WITH NO ADDRESS BITS, OR WE HAVE
2879                                  ;CORE OR THE NEW MOS DESIGN WITH ADDRESS BITS OK!!
2880
2881  011664  026637  000002  001332          CMP     2(SP),@#$BDDAT    ;WAS THE CORRECT PC PUSHED
2882                                                                    ;ON THE STACK?
2883  011672  001404                          BEQ     9$                ;BRANCH IF YES
2884  011674  016637  000002  001334          MOV     2(SP),@#$BDDAT    ;SAVE INCORRECT PC FOR PRINTOUT
2885  011702  000200                          RTS     R0                ;GO BACK TO INDICATE BAD ABORT
2886  011704  005720                  9$:     TST     (R0)+             ;STEP UP RETURN ADDRESS
2887                                                                    ;TO BYPASS THE ERROR HLT
2888  011706  000200                          RTS     R0                ;RETURN TO CONTINUE TESTING
2889  011710  062703  004000          3$:     ADD     #4000,R3          ;STEP UP TO THE NEXT POSSIBLE
2890                                                                    ;ABORT LOCATION AREA
2891  011714  062702  000040                  ADD     #40,R2            ;CHANGE THE VALUE OF THE HIGH
2892                                                                    ;ORDER ADDRESS BITS VALUE TO BE
2893                                                                    ;CHECKED
2894  011720  000742                          BR      1$                ;GO BACK TO CHECK THIS ONE
2895  011722  010237  001326          4$:     MOV     R2,@#$GDAUR       ;STORE WHAT THE ADDRESS BITS
2896                                                                    ;SHOULD HAVE BEEN
2897  011726  010437  001330                  MOV     R4,@#$BDAUR       ;STORE WHAT THE
2898                                                                    ;ADDRESS BITS WERE
2899  011732  012604                          MOV     (SP)+,R4          ;RESTORE R4 CONTENTS
2900  011734  012603                          MOV     (SP)+,R3          ;RESTORE R3 CONTENTS
2901  011736  012602                          MOV     (SP)+,R2          ;RESTORE R2 CONTENTS
```



```
2902
2903                                  ;IF WE HAVE REACHED THIS PATH WE HAVE CORE OR THE NEW MOS DESIGN
2904                                  ;WITH BAD ADDRESS BITS WHICH SHOULD INDICATE AN IMPROPER ABORT!!
2905
2906  011740  026637  000002  001332          CMP     2(SP),@#$BDDAT    ;WAS THE CORRECT PC PUSHED
2907                                                                    ;ON THE STACK?
2908  011746  001404                          BEQ     10$               ;BRANCH IF YES
2909  011750  016637  000002  001334          MOV     2(SP),@#$BDDAT    ;SAVE INCORRECT PC FOR PRINTOUT
2910  011756  000200                          RTS     R0                ;GO BACK TO INDICATE BAD ABORT
2911  011760  016737  167346  001334  10$:     MOV     $GDDAT,@#$BDDAT   ;RECORD THE FACT THAT THE
2912                                                                    ;CORRECT PC WAS PUSHED ON STACK
2913  011766  000200                          RTS     R0                ;GO BACK TO INDICATE BAD ABORT
2914
```

```
                                        ;*****************************************************************
                                        ;END OF PASS ROUTINE
                                        ;INCREMENT THE PASS NUMBER
                                        ;IF SW10=0 DING THE TTY BELL ON END OF PROGRAM
                                        ;IF THERE IS A MONITOR GO TO IT,
                                        ;IF NONE JUMP TO START
011770  000004                  $EOP:   SCOPE
011772  005067  167304                  CLR     $TSTNM                  ;ZERO THE TEST NUMBER
011776  005067  000306                  CLR     $TIMES                  ;ZERO THE NUMBER OF ITERATIONS
012002  005267  167272                  INC     $PASS                   ;INCREMENT THE PASS NUMBER
012006  032737  002000  177570          BIT     #SW10,@#SWR             ;RING THE BELL?
012014  001002                          BNE     4$                      ;NO
012016  104400  012056                  TYPE    ,$BELL                  ;RING A DING
012022  013700  000042          4$:     MOV     @#42,R0                 ;GET MONITOR ADDRESS
012026  001411                          BEQ     $DOAGN                  ;IF NONE
012030  022760  177777  000002          CMP     #-1,2(R0)
012036  001001                          BNE     $ENDAD
012040  000005                          RESET
012042  004710                  $ENDAD: JSR     PC,(R0)                 ;GO TO MONITOR
012044  000240                          NOP                             ;SAVE ROOM
012046  000240                          NOP                             ;FOR
012050  000240                          NOP                             ;ACT11
012052  000137  003620          $DOAGN: JMP     @#START                 ;RETURN
012056  177607  000377          $BELL:  .ASCIZ  <207><377><377>
```



```
                                        ;*****************************************************************
                                        ;THIS ROUTINE IS THE SCOPE HANDLER
                                        ;SW14=1 LOOP ON TEST
                                        ;SW11=1 INHIBIT ITERATIONS
                                        ;SW09=1 LOOP ON ERROR
                                        ;SW08=1 LOOP ON TEST IN SW<7:0>
                                        ;THE TEST NUMBER ($TSTNM) IS UPDATED AND DISPLAYED
                                        $SCOPE:
012062  006137  177570                          ROL     @#SWR                   ;LOOP ON PRESENT TEST?
012066  100502                                  BMI     $OVER                   ;YES IF SW14=1
                                                ;#####START OF CODE FOR THE XOR TESTER#####
012070  000416                  $XTSTR: BR      6$                      ;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;THIS INSTRUCTION TO A "NOP" (NOP=240)
012072  013746  000004                  MOV     @#4,-(SP)               ;SAVE THE CONTENTS OF THE ERROR VECTOR
012076  012737  012116  000004          MOV     #5$,@#4                 ;SET FOR TIMEOUT
012104  005737  177060                  TST     @#177060                ;TIME OUT ON XOR?
012110  012637  000004                  MOV     (SP)+,@#4               ;RESTORE THE ERROR VECTOR
012114  000457                          BR      $SVLAD                  ;GO TO THE NEXT TEST
012116  022626                  5$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIME OUT
012120  012637  000004                  MOV     (SP)+,@#4               ;RESTORE THE ERROR VECTOR
012124  000417                          BR      7$                      ;LOOP ON THE PRESENT TEST
012126                                  ;#####END OF CODE FOR THE XOR TESTER#####
012126  032737  000400  177570          BIT     #SW08,@#SWR             ;LOOP ON SPEC. TEST?
012134  001404                          BEQ     2$                      ;BR IF NO
012136  123767  177570  167136          CMPB    @#SWR,$TSTNM            ;ON THE RIGHT TEST?     SWR<7:0>
012144  001453                          BEQ     $OVER                   ;BR IF YES
012146  105767  167142          2$:     TSTB    $ERFLG                  ;HAS AN ERROR OCCURRED?
012152  001415                          BEQ     3$                      ;BR IF NO
012154  032737  001000  177570          BIT     #SW09,@#SWR             ;LOOP ON ERROR?
012162  001404                          BEQ     4$                      ;BR IF NO
012164  016767  167120  167114  7$:     MOV     $LPERR,$LPADR           ;SET LOOP ADDRESS TO LAST SCOPE
012172  000440                          BR      $OVER
012174  105067  167114          4$:     CLRB    $ERFLG                  ;ZERO THE ERROR FLAG
012200  005067  000104                  CLR     $TIMES                  ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
012204  000415                          BR      1$                      ;ESCAPE TO THE NEXT TEST
012206  032737  004000  177570  3$:     BIT     #SW11,@#SWR             ;INHIBIT ITERATIONS?
012214  001011                          BNE     1$                      ;BR IF YES
012216  005767  167056                  TST     $PASS                   ;IF FIRST PASS OF PROGRAM
012222  001406                          BEQ     1$                      ;       INHIBIT ITERATIONS
012224  005267  167054                  INC     $ICNT                   ;INCREMENT ITERATION COUNT
012230  026767  000054  167046          CMP     $TIMES,$ICNT            ;CHECK THE NUMBER OF ITERATIONS MADE
012236  002016                          BGE     $OVER                   ;BR IF MORE ITERATION REQUIRED
012240  012767  000001  167036  1$:     MOV     #1,$ICNT                ;REINITIALIZE THE ITERATION COUNTER
012246  016767  000040  000034          MOV     $MXCNT,$TIMES           ;SET NUMBER OF ITERATIONS TO DO
012254  005267  167022          $SVLAD: INC     $TSTNM                  ;COUNT TEST NUMBERS
012260  011667  167022                  MOV     (SP),$LPADR             ;SAVE SCOPE LOOP ADDRESS
012264  011667  167020                  MOV     (SP),$LPERR             ;SAVE ERROR LOOP ADDRESS
012270  005067  000500                  CLR     $ESCAPE                 ;CLEAR THE ESCAPE FROM ERROR ADDRESS
012274  016737  167002  177570  $OVER:  MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER
012302  016716  167000                  MOV     $LPADR,(SP)             ;FUDGE RETURN ADDRESS
012306  000002                          RTI                             ;FIXES PS
012310  000000                  $TIMES: 0                               ;NUMBER OF ITERATIONS TO PERFORM
012312  000000                  $MXCNT: 0                               ;MAX. NUMBER OF ITERATIONS
```

```
2993                                    ;*************************************************************
2994                                    ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
2995                                    ;CALL:
2996                                    ;       ACCEPT  ,ADDR           ;PUT OCTAL NUMBER IN ADDR
2997                                    ;
2998  012314                            $ACCEPT:
2999  012314  010046                            MOV     R0,-(SP)                ;PUSH R0 ON STACK
3000  012316  010146                            MOV     R1,-(SP)                ;PUSH R1 ON STACK
3001  012320  010246                            MOV     R2,-(SP)                ;PUSH R2 ON STACK
3002  012322  010346                            MOV     R3,-(SP)                ;PUSH R3 ON STACK
3003  012324  016600  000010                    MOV     10(SP),R0               ;GET ADDRESS POINTER OF WHERE TO PUT NUMBER
3004  012330  005001                    1$:     CLR     R1              ;CLEAR PARTIAL NUMBER
3005  012332  012702  000006                    MOV     #6,R2           ;MAX, # OF DIGITS ALLOWED
3006  012336  104402                    2$:     GETCHR                  ;GET ONE CHARACTER
3007  012340  112603                            MOVB    (SP)+,R3        ;AND PUT IT IN R3
3008  012342  110367  000102                    MOVB    R3,6$
3009  012346  104400  012450                    TYPE    ,6$             ;ECHO THE CHARACTER
3010  012352  042703  000015                    CMP     #15,R3          ;WAS THIS CHARACTER A "CR"?
3011  012356  001422                            BEQ     5$              ;BR IF YES
3012  012360  042703  000060                    CMP     #'0,R3          ;INSURE THE CHARACTER IS
3013  012364  003014                            BGT     4$              ;A DIGIT BETWEEN 0 AND 7.
3014  012366  042703  000067                    CMP     #'7,R3
3015  012372  002411                            BLT     4$
3016  012374  005302                            DEC     R2              ;CHECK NUMBER OF CHARACTERS
3017  012376  002407                            BLT     4$              ;BR IF TO MANY
3018  012400  006301                            ASL     R1              ;POSITION PARTIAL NUMBER
3019  012402  006301                            ASL     R1              ;FOR THIS DIGIT
3020  012404  006301                            ASL     R1
3021  012406  042703  177770                    BIC     #^C<7>,R3       ;GET RID OF THE ASCII JUNK
3022  012412  050301                            BIS     R3,R1           ;COMBINE THIS DIGIT WITH PARTIAL
3023  012414  000750                            BR      2$              ;GO GET ANOTHER DIGIT
3024  012416  104400  012452            4$:     TYPE    ,$QUES          ;TYPE "?"
3025  012422  000742                            BR      1$              ;GO START OVER
3026  012424  104400  012454            5$:     TYPE    ,$LF            ;FOLLOW "CR" WITH A "LF"
3027  012430  010130                            MOV     R1,@(R0)+       ;PASS THE NUMBER TO THE USER
3028  012432  010066  000010                    MOV     R0,10(SP)       ;SET FOR RETURN
3029  012436  012603                            MOV     (SP)+,R3                ;POP STACK INTO R3
3030  012440  012602                            MOV     (SP)+,R2                ;POP STACK INTO R2
3031  012442  012601                            MOV     (SP)+,R1                ;POP STACK INTO R1
3032  012444  012600                            MOV     (SP)+,R0                ;POP STACK INTO R0
3033  012446  000002                            RTI
3034  012450     000     000            6$:     .BYTE   0,0             ;STORAGE FOR ASCII CHAR, AND TERMINATOR
3035  012452     077                    $QUES:  .ASCII  "?"                     ;QUESTION MARK
3036  012453     015                    $CRLF:  .ASCII  <15>                    ;CARRAIGE RETURN
3037  012454  000012                    $LF:    .ASCIZ  <12>                    ;LINEFEED
3038                                                                            ;SELECTED REGISTER FROM THE TTY
3039                                    ;*************************************************************
3040                                    ;THIS ROUTINE INPUTS A SINGLE CHARACTER FROM THE TTY
3041                                    ;CALL:
3042                                    ;
3043                                    ;       GETCHR                  ;INPUT A SINGLE CHARACTER FROM THE TTY
3044                                    ;       RETURN HERE             ;CHARACTER IS ON THE STACK
3045                                    ;
3046  012456  011646                    $GETCHR: MOV    (SP),-(SP)      ;PUSH DOWN THE PC
```



```
3047  012460  016666  000004  000002            MOV     4(SP),2(SP)     ;SAVE THE PS
3048  012466  105777  000126            1$:     TSTB    @TKS            ;WAIT FOR
3049  012472  100375                            BPL     1$              ;A CHARACTER
3050  012474  117766  000122  000004            MOVB    @TKB,4(SP)      ;READ THE TTY
3051  012502  042766  177600  000004            BIC     #^C<177>,4(SP)  ;GET RID OF JUNK IF ANY
3052  012510  000002                            RTI                     ;GO BACK TO USER
3053                                    ;*************************************************************
3054                                    ;THIS ROUTINE INPUTS A STRING FROM THE TTY
3055                                    ;CALL:
3056                                    ;       GETSTR                  ;INPUT A STRING FROM THE TTY
3057                                    ;       RETURN HERE             ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3058                                    ;                               ;TERMINATOR WILL BE A BINARY 0
3059                                    ;
3060  012512  010346                    $GETSTR: MOV    R3,-(SP)        ;SAVE R3
3061  012514  012703  012624            1$:     MOV     #$TTYIN,R3      ;GET ADDRESS
3062  012520  042703  012634            2$:     CMP     #$TTYIN+8.,R3   ;BUFFER FULL?
3063  012524  101405                            BLOS    4$              ;BR IF YES
3064  012526  104402                            GETCHR                  ;GO READ ONE CHARACTER FROM THE TTY
3065  012530  112613                            MOVB    (SP)+,(R3)      ;GET CHARACTER
3066  012532  122713  000177                    CMPB    #177,(R3)       ;IS IT A RUBOUT
3067  012536  001003                            BNE     3$              ;SKIP IF NOT
3068  012540  104400  012452            4$:     TYPE    ,$QUES          ;TYPE A '?'
3069  012544  000763                            BR      1$              ;ZAP THE BUFFER AND LOOP
3070  012546  111367  000044            3$:     MOVB    (R3),8$         ;ECHO THE CHARACTER
3071  012552  104400  012616                    TYPE    ,8$
3072  012556  122723  000015                    CMPB    #15,(R3)+       ;CHECK FOR RETURN
3073  012562  001356                            BNE     2$              ;LOOP IF NOT RETURN
3074  012564  105063  177777                    CLRB    -1(R3)          ;ZAP RETURN (THE 15)
3075  012570  104400  012454                    TYPE    ,$LF            ;TYPE A LINE FEED
3076  012574  012603                            MOV     (SP)+,R3        ;RESTORE R3
3077  012576  011646                            MOV     (SP),-(SP)      ;ADJUST THE STACK AND PUT ADDRESS OF THE
3078  012600  016666  000004  000002            MOV     4(SP),2(SP)     ;       FIRST ASCII CHARACTER ON IT
3079  012606  012766  012624  000004            MOV     #$TTYIN,4(SP)
3080  012614  000002                            RTI                     ;RETURN
3081  012616     000                    8$:     .BYTE   0               ;STORAGE FOR ASCII CHAR, TO TYPE
3082  012617     000                            .BYTE   0               ;TERMINATOR
3083  012620  177560                    TKS:    177560                  ;TTY KBD STATUS
3084  012622  177562                    TKB:    177562                  ;TTY KBD BUFFER
3085  012624  000010                    $TTYIN: .BLKB   8.              ;RESERVE 8 BYTES FOR TTY INPUT
3086                                                                    ;FROM THE TTY
```

```
                                        ;*************************************************************
                                        ;THIS ROUTINE IS THE HLT HANDLER
                                        ;SW09=1 LOOP ON ERROR
                                        ;SW10=1 BELL ON ERROR
                                        ;SW13=1 INHIBIT ERROR TYPEOUTS
                                        ;SW15=1 HALT ON ERROR
                                        ;GO TO TYPERR ON ERROR
012634                                  $HLT:
012634  042777  000001  166760                BIC     #BIT0,@PARITY
012642  112767  000001  166444                MOVB    #1,$ERFLG               ;SET THE ERROR FLAG
012650  032737  002000  177570                BIT     #SW10,@#SWR             ;BELL ON ERROR?
012656  001402                                BEQ     1$                      ;NO - SKIP
012660  104400  012056                        TYPE    ,$BELL                  ;RING BELL
012664  005267  166422                1$:     INC     $ERTTL                  ;COUNT THE NUMBER OF ERRORS
012670  001775                                BEQ     1$                      ;INSURE $ERTTL NOT=0
012672  011667  166426                        MOV     (SP),$HLTAD             ;GET ADDRESS OF HLT INSTRUCTION
012676  162767  000002  166420                SUB     #2,$HLTAD
012704  117767  166414  166410                MOVB    @$HLTAD,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
012712  032737  020000  177570                BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
012720  001004                                BNE     2$                      ;SKIP TYPEOUTS
012722  004737  012776                        JSR     PC,@#TYPERR             ;GO TO USER ERROR ROUTINE
012726  104400  012453                        TYPE    ,$CRLF
012732  005737  177570                2$:     TST     @#SWR                   ;HALT ON ERROR
012736  100001                                BPL     3$                      ;SKIP IF CONTINUE
012740  000000                                HALT                            ;HALT ON ERROR!
012742  032737  001000  177570        3$:     BIT     #SW09,@#SWR             ;LOOP ON ERROR SWITCH SET?
012750  001403                                BEQ     4$                      ;BR IF NO
012752  016716  166332                        MOV     $LPERR,(SP)             ;FUDGE RETURN FOR LOOPING
012756  000002                                RTI
012760  005767  000010                4$:     TST     $ESCAPE                 ;CHECK FOR AN ESCAPE ADDRESS
012764  001402                                BEQ     5$                      ;BR IF NONE--TAKE NORMAL RETURN
012766  016716  000002                        MOV     $ESCAPE,(SP)            ;FUDGE RETURN ADDRESS FOR ESCAPE
012772  000002                        5$:     RTI
012774  000000                        $ESCAPE:        0                       ;ESCAPE ON ERROR ADDRESS
                                                                            ;MESSAGE TYPEOUT ROUTINE
                                                                            ;DEFINED BY 'TYPERR' AND
                                                                            ;1ST INSTRUCTION OF ERROR 'HLT'
                                                                            ;ROUTINE
                                        ;*************************************************************
                                        ;
                                        ;THIS ROUTINE WILL TYPE OUT THE ERROR MESSAGES
                                        ;
                                        ;*************************************************************
012776  104400  012453                TYPERR: TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN
                                                                            ;AND LINE FEED
013002  010046                                MOV     R0,-(SP)                ;SAVE R0 CONTENTS
013004  005000                                CLR     R0                      ;CLEAR R0
013006  116700  166310                        MOVB    $ITEMB,R0               ;PICK UP THE ITEM INDEX
013012  005300                                DEC     R0                      ;ADJUST THE INDEX
013014  006300                                ASL     R0                      ;SO IT WILL WORK FOR
013016  006300                                ASL     R0                      ;FOR THE ERROR TABLE
013020  006300                                ASL     R0
013022  062700  001500                        ADD     #$ERRTB,R0              ;FORM THE TABLE POINTER
```



```
013026  012067  000002                        MOV     (R0)+,1$                ;PICK UP 'ERROR MESSAGE' POINTER
013032  104400                                TYPE                            ;TYPE 'ERROR MESSAGE'
013034  000000                        1$:     0                               ;'ERROR MESSAGE' POINTER GOES HERE
013036  104400  012453                        TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN AND
                                                                            ;LINE FEED
013042  012067  000004                        MOV     (R0)+,2$                ;PICK UP 'DATA HEADER' POINTER
013046  001402                                BEQ     3$                      ;IF '0' DON'T TYPE
013050  104400                                TYPE                            ;TYPE 'DATA HEADER'
013052  000000                        2$:     0                               ;'DATA HEADER' POINTER GOES HERE
013054  104400  012453                3$:     TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN AND
                                                                            ;LINE FEED
013060  012000                                MOV     (R0)+,R0                ;PICK UP 'DATA POINTER'
013062  001004                                BNE     5$                      ;IF THERE IS DATA TO TYPE GO DO IT
013064  012600                        4$:     MOV     (SP)+,R0                ;RESTORE R0
013066  104400  012453                        TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN AND
                                                                            ;AND LINE FEED
013072  000207                                RTS     PC                      ;RETURN TO TESTING
013074                                  5$:
013074  013046                                MOV     @(R0)+,-(SP)            ;SAVE @(R0)+ FOR TYPEOUT
                                                                            ;TYPE DATA
013076  004067  000246                        JSR     R0,$B2OCT               ;GO TYPE-OCTAL ASCII
013102     006                                .BYTE   6                       ;TYPE 6 DIGITS
013103     001                                .BYTE   1                       ;TYPE LEADING ZEROS
013104  005710                                TST     (R0)                    ;HAVE WE REACHED THE '0' TERMINATOR
013106  001766                                BEQ     4$                      ;YES - CLEAN UP FOR RETURN
013110  104400  013116                        TYPE    ,6$                     ;TYPE 5 SPACES
013114  000767                                BR      5$                      ;LOOP TILL '0' TERMINATOR REACHED
013116  040040  020040  000040        6$:     .ASCIZ  /     /
                                              .EVEN
```

```
3170                                    ;*****************************************************************
3171                                    ;
3172                                    ;THE FOLLOWING ROUTINE WILL SIZE MEMORY
3173                                    ;
3174                                    ;$LSTBLK WILL CONTAIN THE LAST BANK AS AN SAF (SEGMENT ADDRESS FIELD)
3175                                    ;
3176                                    ;THE SEGMENT ADDRESS FIELD CONTAINS THE 4 MOST SIGNIFICANT BITS OF
3177                                    ;THE LAST ADDRESS OF THE LAST BANK FOUND
3178                                    ;
3179                                    ;*****************************************************************
3180  013124  010546                   $SIZE:  MOV     R5,-(SP)                ;SAVE R5 CONTENTS
3181  013126  010605                           MOV     SP,R5                   ;SAVE THE STACK POINTER
3182  013130  012703  077406                   MOV     #77406,R3
3183  013134  010377  165050                   MOV     R3,@KPDR0               ;SET THE FOLLOWING PAGE
3184  013140  010377  165046                   MOV     R3,@KPDR1               ;DESCRIPTOR REGISTERS TO
3185  013144  010377  165044                   MOV     R3,@KPDR2               ;READ/WRITE AND TRANSFER OF
3186  013150  010377  165042                   MOV     R3,@KPDR3               ;4096 (10) WORDS PER SEGMENT
3187  013154  010377  165040                   MOV     R3,@KPDR4
3188  013160  010377  165036                   MOV     R3,@KPDR5
3189  013164  010377  165034                   MOV     R3,@KPDR6
3190  013170  010377  165032                   MOV     R3,@KPDR7
3191  013174  005077  165030                   CLR     @KPAR0                  ;SET THE FOLLOWING PAGE
3192  013200  012777  000200  165024           MOV     #200,@KPAR1             ;ADDRESS REGISTERS TO THEIR
3193  013206  012777  000400  165020           MOV     #400,@KPAR2             ;RESPECTIVE OFFSET VALUES
3194  013214  012777  000600  165014           MOV     #600,@KPAR3             ;FOR RELOCATION PURPOSES
3195  013222  012777  001000  165010           MOV     #1000,@KPAR4
3196  013230  012777  001200  165004           MOV     #1200,@KPAR5
3197  013236  012777  001400  165000           MOV     #1400,@KPAR6
3198  013244  012777  007600  164774           MOV     #7600,@KPAR7            ;THIS ONE'S THE I/O RECORD
3199                                                                           ;PAGE CONTAINING CONTROL STATUS
3200                                                                           ;REGISTERS, ETC.
3201  013252  016704  164766                   MOV     KPAR6,R4                ;GET ADDRESS OF PAGE 6 REGISTER
3202  013256  005014                           CLR     (R4)                    ;CLEAR THE REGISTER
3203  013260  005277  164720                   INC     @SR0                    ;TURN ON MEMORY MANAGEMENT
3204  013264  010746                           MOV     PC,-(SP)                ;MAKE KT11 TIMEOUT SERVICE
3205  013266  062716  000026                   ADD     #$KTOUT-.,(SP)          ;ROUTINE ADDRESS POSITION
3206                                                                           ;INDEPENDENT
3207  013272  012637  000004                   MOV     (SP)+,@#ERRVEC          ;SET FOR TIMEOUT
3208  013276  005737  143776           1$:     TST     @#143776                ;TRAP ON NON-EXISTENT MEMORY
3209  013302  062714  000040                   ADD     #40,(R4)                ;MAKE A 1K STEP
3210  013306  027714  164734                   CMP     @KPAR7,(R4)             ;LAST ONE?
3211  013312  003371                           BGT     1$                      ;NO - TRY IT!
3212  013314  011400                   $KTOUT: MOV     (R4),R0                 ;GET LAST BANK +1
3213  013316  162700  000040                   SUB     #40,R0                  ;DROP BACK
3214  013322  012737  000006  000004           MOV     #6,@#ERRVEC             ;SET FOR ERRORS
3215  013330  010506                           MOV     R5,SP                   ;RESTORE THE STACK POINTER
3216  013332  010067  000010                   MOV     R0,$LSTBLK              ;STORE THE SAF
3217  013336  005077  164642                   CLR     @SR0                    ;TURN MEMORY MGMT OFF
3218  013342  012605                   1$:     MOV     (SP)+,R5                ;RESTORE R5
3219  013344  000207                           RTS     PC                      ;RETURN TO NORMAL FLOW
3220  013346  000000                   $LSTBLK:        .WORD   0               ;CONTAINS THE SAF
3221                                    ;*****************************************************************
3222                                    ;BINARY TO OCTAL (ASCII) AND TYPE
3223                                    ;$B2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
```



```
3224                                    ;CALL:
3225                                    ;       MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
3226                                    ;       JSR     R0,$B2OCT               ;CALL FOR TYPEOUT
3227                                    ;       .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3228                                    ;       .BYTE   M                       ;M=1 OR 0
3229                                    ;                                               ;1=TYPE LEADING ZEROS
3230                                    ;                                               ;0=SUPPRESS LEADING ZEROS
3231                                    ;
3232                                    ;$B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B2OCT OR $B2016
3233                                    ;CALL:
3234                                    ;       MOV     NUM,-(SP)
3235                                    ;       JSR     R0,$B201
3236                                    ;
3237                                    ;$B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3238                                    ;CALL:
3239                                    ;       MOV     NUM,-(SP)
3240                                    ;       JSR     R0,$B2016
3241
3242  013350  112067  000201           $B2OCT: MOVB    (R0)+,$OMODE+1          ;PICKUP THE NUMBER OF DIGITS TO TYPE
3243  013354  112067  000173                   MOVB    (R0)+,$0FILL            ;GET THE ZERO FILL SWITCH
3244  013360  000406                           BR      $B201
3245  013362  112767  000001  000163   $B2016: MOVB    #1,$0FILL               ;SET THE ZERO FILL SWITCH
3246  013370  112767  000006  000157           MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
3247  013376  112767  000005  000146   $B201:  MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
3248  013404  010346                           MOV     R3,-(SP)                ;SAVE R3
3249  013406  010446                           MOV     R4,-(SP)                ;SAVE R4
3250  013410  010546                           MOV     R5,-(SP)                ;SAVE R5
3251  013412  116704  000137                   MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
3252  013416  005404                           NEG     R4
3253  013420  062704  000006                   ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
3254  013424  110467  000124                   MOVB    R4,$OMODE               ;SAVE IT FOR USE
3255  013430  116704  000117                   MOVB    $0FILL,R4               ;GET THE ZERO FILL SWITCH
3256  013434  016605  000010                   MOV     10(SP),R5               ;PICKUP THE INPUT NUMBER
3257  013440  005003                           CLR     R3                      ;CLEAR THE OUTPUT WORD
3258  013442  006105                   1$:     ROL     R5                      ;ROTATE MSB INTO "C"
3259  013444  000404                           BR      3$                      ;GO DO MSB
3260  013446  006105                   2$:     ROL     R5                      ;FORM THIS DIGIT
3261  013450  006105                           ROL     R5
3262  013452  006105                           ROL     R5
3263  013454  010503                           MOV     R5,R3
3264  013456  006103                   3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
3265  013460  105367  000070                   DECB    $OMODE                  ;TYPE THIS DIGIT?
3266  013464  100016                           BPL     7$                      ;BR IF NO
3267  013466  042703  177770                   BIC     #177770,R3              ;GET RID OF JUNK
3268  013472  001002                           BNE     4$                      ;TEST FOR 0
3269  013474  005704                           TST     R4                      ;SUPPRESS THIS 0?
3270  013476  001403                           BEQ     5$                      ;BR IF YES
3271  013500  005204                   4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
3272  013502  052703  000060                   BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
3273  013506  052703  000040           5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
3274  013512  110367  000032                   MOVB    R3,8$                   ;SAVE FOR TYPING
3275  013516  104400  013550                   TYPE    ,8$                     ;GO TYPE THIS DIGIT
3276  013522  105367  000024           7$:     DECB    $OCNT                   ;COUNT BY 1
3277  013526  003347                           BGT     2$                      ;BR IF MORE TO DO
```

```
3278  013530  002402                        BLT     6$                      ;BR IF DONE
3279  013532  005204                        INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
3280  013534  000744                        BR      2$                      ;GO DO THE LAST DIGIT
3281  013536  012605                6$:     MOV     (SP)+,R5                ;RESTORE R5
3282  013540  012604                        MOV     (SP)+,R4                ;RESTORE R4
3283  013542  012603                        MOV     (SP)+,R3                ;RESTORE R3
3284  013544  012616                        MOV     (SP)+,(SP)              ;SET THE STACK FOR RETURNING
3285  013546  000200                        RTS     R0                      ;RETURN
3286  013550     000                8$:     .BYTE   0                       ;STORAGE FOR ASCII DIGIT
3287  013551     000                        .BYTE   0                       ;TERMINATOR FOR TYPE ROUTINE
3288  013552     000                $OCNT:  .BYTE   0                       ;OCTAL DIGIT COUNTER
3289  013553     000                $OFILL: .BYTE   0                       ;ZERO FILL SWITCH
3290  013554  000000                $OMODE: 0                               ;NUMBER OF DIGITS TO TYPE
```



```
3291
3292                                    ;*****************************************************************
3293                                    ;TRAP HANDLER
3294  013556  010046                    $TRAP:  MOV     R0,-(SP)        ;SAVE R0
3295  013560  016600  000002                    MOV     2(SP),R0        ;GET TRAP ADDRESS
3296  013564  005740                            TST     -(R0)           ;BACKUP BY 2
3297  013566  111000                            MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP
3298  013570  016000  013576                    MOV     $TRPAD(R0),R0   ;INDEX TO TABLE
3299  013574  000200                            RTS     R0              ;GO TO ROUTINE
3300
3301
3302                                    ;TRAP TABLE
3303                                    ;       ROUTINE
3304                                    ;       -------
3305  013576  001110                    $TRPAD: $TYPE   ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
3306  013600  012456                            $READC  ;CALL=GETCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
3307  013602  012512                            $READS  ;CALL=GETSTR    TRAP+4(104404)  TTY TYPEIN STRING ROUTINE
3308  013604  012314                            $ACCEPT ;CALL=ACCEPT    TRAP+6(104406)  READ AN OCTAL NUMBER FROM TTY
```

```
                                        ;*****************************************************************
                                        ;POWER DOWN ROUTINE
013606  012737  013734  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC        ;SET FOR FAST UP
013614  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;PRIO:7
013622  010046                          MOV     R0,-(SP)                ;PUSH R0 ON STACK
013624  010146                          MOV     R1,-(SP)                ;PUSH R1 ON STACK
013626  010246                          MOV     R2,-(SP)                ;PUSH R2 ON STACK
013630  010346                          MOV     R3,-(SP)                ;PUSH R3 ON STACK
013632  010446                          MOV     R4,-(SP)                ;PUSH R4 ON STACK
013634  010546                          MOV     R5,-(SP)                ;PUSH R5 ON STACK
013636  010667  000076                  MOV     SP,$SAVR6               ;SAVE SP
013642  012737  013654  000024          MOV     #$PWRUP,@#PWRVEC        ;SET UP VECTOR
013650  000000                          HALT
013652  000776                          BR      .-2                     ;HANG UP
                                        ;*****************************************************************
                                        ;POWER UP ROUTINE
013654  016706  000060  $PWRUP: MOV     $SAVR6,SP               ;GET SP
013660  005067  000054          CLR     $SAVR6                  ;WAIT LOOP FOR THE TTY
013664  005267  000050  1$:     INC     $SAVR6                  ;WAIT FOR THE INC
013670  001375                  BNE     1$                      ;OF  WORD
013672  012605                  MOV     (SP)+,R5                ;POP STACK INTO R5
013674  012604                  MOV     (SP)+,R4                ;POP STACK INTO R4
013676  012603                  MOV     (SP)+,R3                ;POP STACK INTO R3
013700  012602                  MOV     (SP)+,R2                ;POP STACK INTO R2
013702  012601                  MOV     (SP)+,R1                ;POP STACK INTO R1
013704  012600                  MOV     (SP)+,R0                ;POP STACK INTO R0
013706  012737  013606  000024  $PWRVEC: MOV     #$PWRDN,@#PWRVEC        ;SET UP THE POWER DOWN VECTOR
013714  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;PRIO:7
013722  104400  013742                  TYPE    ,$POWER                 ;POWER FAIL MESSAGE
013726  012716  001706                  MOV     #BEGIN,(SP)             ;RESTART AT BEGIN
013732  000002                          RTI
013734  000000  $ILLUP: HALT                                    ;THE POWER UP SEQUENCE WAS STARTED
013736  000776          BR      .-2                             ; BEFORE THE POWER DOWN WAS COMPLETE
013740  000000  $SAVR6: 0                                       ;PUT THE SP HERE
013742  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
013750  000122
                                        .EVEN
```



```
                                        ;*****************************************************************
                                        ;
                                        ;ERROR AND MESSAGE TABLE CONDIMENTS
                                        ;
                                        ;*****************************************************************

013752  042524  052123  042040  EM1:    .ASCIZ  /TEST DIDN'T ABORT  /
013760  042111  023516  020124
013766  041101  051117  020124
013774  000040
013776  040506  040524  020114  EM2:    .ASCIZ  /FATAL ERROR TO PROGRAM  /
014004  051105  047522  020122
014012  047524  050040  047522
014020  051107  046501  020040
014026     000
014027     101  047502  052122  EM3:    .ASCIZ  /ABORTED INCORRECTLY  /
014034  042105  044440  041516
014042  051117  042522  052103
014050  053514  020040     000
014055     116  020117  040520  EM4:    .ASCIZ  /NO PARITY MEMORY FOUND BELOW 28K  /
014062  044522  054524  046440
014070  046505  051117  020131
014076  047506  047125  020104
014104  042502  047514  020127
014112  034062  020113  000040
014120  042522  042523  020124  EM5:    .ASCIZ  /RESET DOESN'T WORK  /
014126  047504  051505  023516
014134  040124  047527  045522
014142  040040     000
014145     125  042523  020122  EM6:    .ASCIZ  /USER SELECTED REGISTER NOT PRESENT  /
014152  042523  042514  052103
014160  042105  051040  043505
014166  051511  042524  020122
014174  047516  020124  051120
014202  051505  047105  020124
014210  000040
014212  047516  050040  051101  EM7:    .ASCIZ  /NO PARITY MEMORY FOUND AT ALL  /
014220  052111  020131  042515
014226  047515  054522  043040
014234  052517  042116  040440
014242  040124  046101  020114
014250  000040
014252  044504  047104  052047  EM10:   .ASCIZ  /DIDN'T ABORT OR RECOGNIZE STACK VIOLATION  /
014260  040440  047502  052122
014266  047440  020122  042522
014274  047503  047107  055111
014302  040105  052123  041501
014310  040113  044526  046117
014316  052101  047511  020116
014324  000040
014326  041101  051117  042524  EM11:   .ASCIZ  /ABORTED BUT STACK VIOLATION NOT RECOGNIZED  /
014334  040104  052502  020124
014342  052123  041501  020113
014350  044526  046117  052101
```

```
3401  014356  047511  020116  047516
3402  014364  042124  042522  047503
3403  014372  047107  055111  042105
3404  014400  042040     000
3405  014403     123  040524  045503  EM12:   .ASCIZ  /STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED  /
3406  014410  053040  047511  040514
3407  014416  044524  047117  050040
3408  014424  041511  042513  020104
3409  014432  050125  041040  052125
3410  014440  040440  047502  052122
3411  014446  047040  052117  051040
3412  014454  041505  043517  044516
3413  014462  042532  020104  000040
3414  014470  051120  043517  040522  DM1:    .ASCII  /PROGRAM   REGISTER/<15><12>
3415  014476  042115  020040  042522
3416  014504  044507  052123  051105
3417  014512  005015
3418  014514  042040  041520  020040          .ASCIZ  /  PC     UNDER TEST/
3419  014522  042040  052440  042116
3420  014530  051105  052040  051505
3421  014536  000124
3422  014540  051040  043505  051511  DM2:    .ASCII  / REGISTER/<15><12>                    #"
3423  014546  042524  006522     012
3424  014553     125  042116  051105          .ASCIZ  /UNDER TEST/
3425  014560  052040  051505  000124
3426  014566  051120  043517  040522  DM3:    .ASCII  /PROGRAM/<15><12>
3427  014574  006515     012
3428  014577     040  050040  000103          .ASCIZ  /  PC/
3429  014604  051120  043517  040522  DM4:    .ASCII  /PROGRAM   REGISTER    EXPECTED      ACTUAL    EXPECTED   ACTUAL/<15><12>
3430  014612  042115  020040  042522
3431  014620  044507  052123  051105
3432  014626  042040  020040  054105
3433  014634  042520  052103  042105
3434  014642  042040  020040  040440
3435  014650  052103  040525  020114
3436  014656  042040  042440  050130
3437  014664  041505  042524  020104
3438  014672  042040  041501  052524
3439  014700  046101  005015
3440  014704  042040  041520  020040          .ASCIZ  /  PC     UNDER TEST  ADDR, BITS  ADDR, BITS  ABORT PC   ABORT PC/
3441  014712  042040  052440  042116
3442  014720  051105  052040  051505
3443  014726  042124  040440  042104
3444  014734  042722  041040  052111
3445  014742  042123  040440  042104
3446  014750  042722  041040  052111
3447  014756  042123  040440  047502
3448  014764  052122  050040  020103
3449  014772  042040  041101  051117
3450  015000  042124  041520     000
3451  015005     120  047522  051107  DM5:    .ASCII  /PROGRAM   REGISTER    EXPECTED/<15><12>
3452  015012  046501  020040  051040
3453  015020  043505  051511  042524
3454  015026  042122  020040  042440
```



```
3455  015034  050130  041505  042524
3456  015042  006504     012
3457  015045     040  050040  020103          .ASCIZ  /  PC     UNDER TEST   ABORT PC/
3458  015052  042040  020040  052440
3459  015060  042116  051105  052040
3460  015066  051505  020124  020040
3461  015074  041101  051117  020124
3462  015102  041520     000
3463          015106                          .EVEN
3464  015106  001324  001622  000000  DI1:    .WORD   $HLTAD,PARITY,0
3465  015114  001622  000000          DI2:    .WORD   PARITY,0
3466  015120  001324  000000          DI3:    .WORD   $HLTAD,0
3467  015124  001324  001622  001326  DI4:    .WORD   $HLTAD,PARITY,$GDADR,$BDADR,$GDDAT,$BDDAT,0
3468  015132  001330  001332  001334
3469  015140  000000
3470  015142  001324  001622  001332  DI5:    .WORD   $HLTAD,PARITY,$GDDAT,0
3471  015150  000000
3472
3473          000001                  .END
```

```
A       003752          ABORT   003430          ACCEPT= 104406          A0      004026
A1      004104          A2      004162          B       004236          BEGIN   001706
BIT0    = 000001        BIT00   = 000001        BIT01   = 000002        BIT02   = 000004
BIT03   = 000010        BIT04   = 000020        BIT05   = 000040        BIT06   = 000100
BIT07   = 000200        BIT08   = 000400        BIT09   = 001000        BIT1    = 000002
BIT10   = 002000        BIT11   = 004000        BIT12   = 010000        BIT13   = 020000
BIT14   = 040000        BIT15   = 100000        BIT2    = 000004        BIT3    = 000010
BIT4    = 000020        BIT5    = 000040        BIT6    = 000100        BIT7    = 000200
BIT8    = 000400        BIT9    = 001000        BLKCNT  001632          BPTVEC= 000014
B0      004312          B1      004370          B2      004444          B3      004536
B4      004612          B5      004706          B6      004766          C       005042
CHECKL  011550          COMPUT  003204          CPU40   001642          C0      005116
C1      005210          C2      005304          C3      005360          C4      005436
C5      005514          C6      005570          C7      005644          C8      005742
D       006020          DD      006006          DD0     006064          DD1     006144
DH1     014470          DH2     014540          DH3     014566          DH4     014604
DH5     015005          DISPLA= 177570          DT1     015106          DT2     015114
DT3     015120          DT4     015124          DT5     015142          D0      006076
D1      006156          E       006320          EMTVEC= 000030          EM1     013752
EM10    014252          EM11    014326          EM12    014403          EM2     013776
EM3     014327          EM4     014055          EM5     014120          EM6     014145
EM7     014212          ERRVEC= 000004          E0      006326          E1      006502
E2      006510          F       006634          FINDON  002504          FLAGSC  011474
F0      006644          F1      006774          G       007056          GETCHR= 104402
GETSTR= 104404          GO      002326          G0      007066          H       007140
HOLDLO  010302          H0      007150          H1      007134          H2      007242
H3      007216          H4      007222          INITIA  011506          INTERT  001644
INTVEC  001620          IOTVEC= 000020          KPAR0   000230          KPAR1   000232
KPAR2   000234          KPAR3   000236          KPAR4   000240          KPAR5   000242
KPAR6   000244          KPAR7   000246          KPDR0   000210          KPDR1   000212
KPDR2   000214          KPDR3   000216          KPDR4   000220          KPDR5   000222
KPDR6   000224          KPDR7   000226          KTTIME  002324          L       007320
LEAFCN  001636          M       007374          MEMAD   001640          MRK0    006324
MRK1    006506          MSGTYP  002372          MSREGF  001626          N       = 000067
NEWSTK  001476          NEXT1   002514          NN      007450          NOMORE  003172
NOREG   003144          NTERAD  001446          NTER0   001450          NTER1   001452
NTER10  001470          NTER11  001472          NTER12  001474          NTER2   001454
NTER3   001456          NTER4   001460          NTER5   001462          NTER6   001464
NTER7   001466          ONETRY  003500          P       007526          PARCOR  003112
PARITY  001622          PARTST  003222          PC      =%000007        PS      = 177776
PSPCOR  001624          PSW     = 177776        PWRVEC= 000024          R       007604
RED     010400          RESTAR  011464          RESTOR  001634          RESVEC= 000010
R0      =%000000        R1      =%000001        R2      =%000002        R3      =%000003
R4      =%000004        R5      =%000005        R6      =%000006        R7      =%000007
SAVLOC  010126          SEGVEC  000250          SP      =%000006        SR0     000204
SR2     000206          STACK   = 001100        START   003620          SWR     = 177570
SW0     = 000001        SW00    = 000001        SW01    = 000002        SW02    = 000004
SW03    = 000010        SW04    = 000020        SW05    = 000040        SW06    = 000100
SW07    = 000200        SW08    = 000400        SW09    = 001000        SW1     = 000002
SW10    = 002000        SW11    = 004000        SW12    = 010000        SW13    = 020000
SW14    = 040000        SW15    = 100000        SW2     = 000004        SW3     = 000010
SW4     = 000020        SW5     = 000040        SW6     = 000100        SW7     = 000200
SW8     = 000400        SW9     = 001000        T       007750          TBITVE= 000014
TKB     012622          TKS     012620          TRAPVE= 000034          TRTVEC= 000014
```



```
TST1    003226          TST10   004046          TST11   004124          TST12   004202
TST13   004256          TST14   004332          TST15   004410          TST16   004464
TST17   004556          TST2    003250          TST20   004632          TST21   004726
TST22   005006          TST23   005062          TST24   005136          TST25   005230
TST26   005324          TST27   005400          TST3    003272          TST30   005456
TST31   005534          TST32   005610          TST33   005664          TST34   005762
TST35   006040          TST36   006116          TST37   006176          TST4    003334
TST40   006352          TST41   006534          TST42   006664          TST43   007014
TST44   007106          TST45   007170          TST46   007262          TST47   007340
TST5    003374          TST50   007414          TST51   007470          TST52   007546
TST53   007624          TST54   007674          TST55   007774          TST56   010076
TST57   010252          TST6    003716          TST60   010426          TST61   010534
TST62   010636          TST63   010740          TST64   011076          TST65   011234
TST66   011372          TST7    003772          TYPE    = 104400        TYPERR  012776
T0      007754          USERTY  001630          V       010052          VECSET  011450
V0      010056          V1      010162          V2      010166          V3      010336
V4      010342          W       010514          WHICH1  003376          XX      010616
Y       010720          YELLOW  010224          Z0      011056          Z1      011214
Z2      011352          $ACCEP  012314          $BDADR  001530          $BDDAT  001334
$BELL   012056          $B20CT  013350          $B201   013376          $B2016  013362
$CM1    = 000013        $CM2    = 000026        $CM3    = 000013        $CM4    = 000013
$CRLF   012453          $DOAGN  012052          $ENDAD  012042          $EOP    011770
$ERFLG  001314          $ERRTB  001500          $ERTTL  001312          $ESCAP  012774
$FILLS  001105          $GDADR  001326          $GDDAT  001332          $HLT    012634
$HLTAD  001324          $ICNT   001304          $ILLUP  013734          $ITEMB  001322
$KTOUT  013314          $KT11   002304          $LF     012454          $LPADR  001306
$LPERR  001310          $LSTBL  013346          $MXCNT  012312          $NULL   001104
$OCNT   013552          $OMODE  013554          $OVER   012274          $PASS   001300
$POWER  013742          $PWRDN  013606          $PWRUP  013654          $QUES   012452
$RDSZ   = 000010        $READC  012456          $READS  012512          $REGAD  001336
$REG0   001340          $REG1   001342          $REG10  001360          $REG11  001362
$REG12  001364          $REG2   001344          $REG3   001346          $REG4   001350
$REG5   001352          $REG6   001354          $REG7   001356          $SAVR6  013740
$SCOPE  012062          $SETAD  001416          $SETUP= 000017          $SET0   001420
$SET1   001422          $SET10  001440          $SET11  001442          $SET12  001444
$SET2   001424          $SET3   001426          $SET4   001430          $SET5   001432
$SET6   001434          $SET7   001436          $SIZE   013124          $SS     = 000001
$STUP   = 177777        $SVLAD  012254          $SWR    = 167400        $TIMES  012310
$TMPAD  001414          $TMP0   001366          $TMP1   001370          $TMP10  001406
$TMP11  001410          $TMP12  001412          $TMP2   001372          $TMP3   001374
$TMP4   001376          $TMP5   001400          $TMP6   001402          $TMP7   001404
$TN     = 000001        $TPB    001102          $TPFLG  001106          $TPS    001100
$TRAP   013556          $TRP    = 000010        $TRPAD  013576          $TSTNM  001302
$TTYIN  012624          $TYPE   001110          $XTSTR  012070          $0FILL  013553
.       = 015152
```

```
 ERRORS DETECTED:  0

 *DCKBRE,DCKBRE/SUL=DCKBRE/DS:ERFZ
 RUN-TIME:  37  30  0 SECONDS
 CORE USED:  14K
```