11/40/45

CPU PARITY TEST

MD-11-DCKBR-E

EP DCKBR E DL A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA

B01

```
$R6578901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012LPT
DBSTART*AGer DAVIES,TOM[4OBEEEEH] SSSS DCKBRE STATTS Da0013-Oct-76 14:57:05 Monitor IPC-D 507B [1A3] *START*
D  D  A   A  V   V   I   E     S           T     O   O  MM MM
D  D  A   A  V   V   I   E     S           T     O   O  M M M
D  D  A   A  V   V   I   EEEE   SSS        T     O   O  M   M
D  D  AAAAA  V   V   I   E         S       T     O   O  M   M
D  D  A   A   V V    I   E         S       T     O   O  M   M
DDDD  A   A    V    III  EEEEE SSSS   ;    T      OOO   M   M


MTO:ML21-4:DAVIE82-- distribution to ML21-4, slot 134    1    333   4  4     >
MM MM  L       2   2    11           4 4       <     11   3   3  4  4      >
M M M  L           2     1           4 4      <       1       3  4  4       >
M   M  L          2      1    -----  44444   <        1      3   44444       >
M   M  L         2       1               4    <       1       3     4       >
M   M  L        2        1               4     <      1   3   3     4      >
M   M  LLLLL  22222     111              4      <    111   333      4     >


/SOMEONE-4:DAVIES,TOM[4OBEEEEH]aJML21[illegible]eSa345 Date 13-Oct-76 14:57:05 Monitor IPC-D 507B [1A3] *START*
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111111111111111111111110
000000000111111111122222222223333333333444444444455555555556666666666777777777788888888889999999999000000000011111111112222222222223331
3456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 *S
D  D  A   A  V   V   I   E     S           T     O   O  MM MM
D  D  A   A  V   V   I   E     S           T     O   O  M M M
D  D  A   A  V   V   I   EEEE   SSS        T     O   O  M   M
D  D  AAAAA  V   V   I   E         S       T     O   O  M   M
D  D  A   A   V V    I   E         S       T     O   O  M   M
DDDD  A   A    V    III  EEEEE SSSS   ;    T      OOO   M   M


MTO:ML21-4:DAVIE82-- distribution to ML21-4, slot 134    1    333   4  4     >
MM MM  L       2   2    11           4 4       <     11   3   3  4  4      >
M M M  L           2     1           4 4      <       1       3  4  4       >
M   M  L          2      1    -----  44444   <        1      3   44444       >
M   M  L         2       1               4    <       1       3     4       >
M   M  L        2        1               4     <      1   3   3     4      >
M   M  LLLLL  22222     111              4      <    111   333      4     >


/SOMEONE-4:DAVIES,TOM[4OBEEEEH]aJML21[illegible]eSa345 Date 13-Oct-76 14:57:05 Monitor IPC-D 507B [1A3] *START*
```

C01

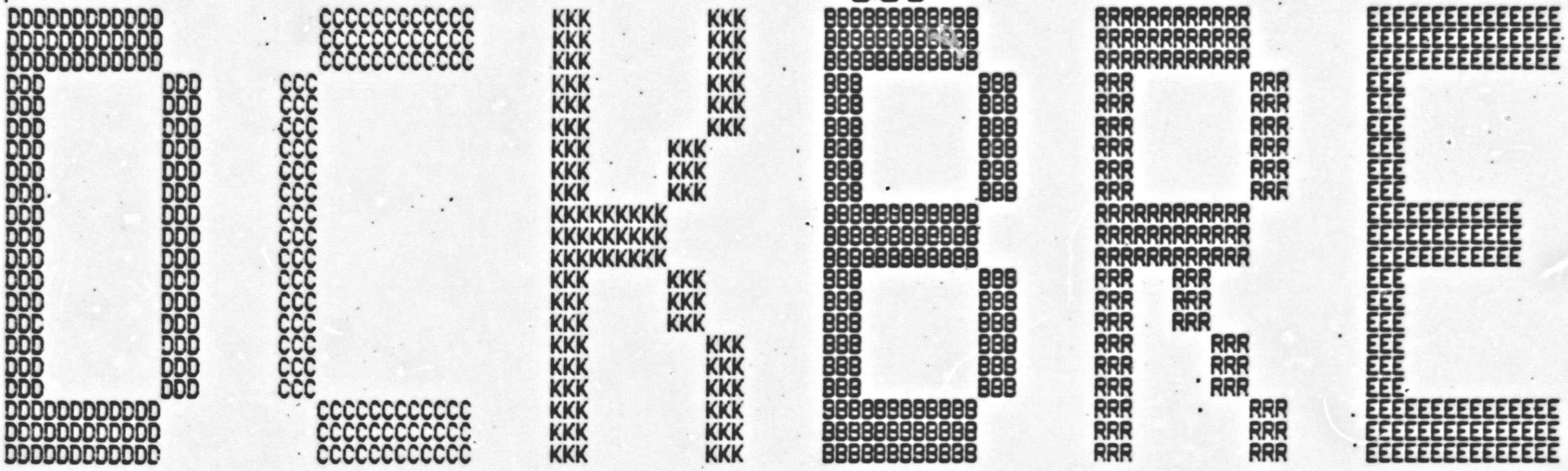
```
DDDDDDDDDDDD            CCCCCCCCCCCC      KKK         KKK      BBBBBBBBBBBB         RRRRRRRRRRRR         EEEEEEEEEEEEEEE
DDDDDDDDDDDD            CCCCCCCCCCCC      KKK         KKK      BBBBBBBBBBBB         RRRRRRRRRRRR         EEEEEEEEEEEEEEE
DDDDDDDDDDDD            CCCCCCCCCCCC      KKK         KKK      BBBBBBBBBBBB         RRRRRRRRRRRR         EEEEEEEEEEEEEEE
DDD         DDD      CCC                  KKK         KKK      BBB         BBB      RRR         RRR      EEE
DDD         DDD      CCC                  KKK         KKK      BBB         BBB      RRR         RRR      EEE
DDD         DDD      CCC                  KKK         KKK      BBB         BBB      RRR         RRR      EEE
DDD         DDD      CCC                  KKK      KKK         BBB         BBB      RRR         RRR      EEE
DDD         DDD      CCC                  KKK      KKK         BBB         BBB      RRR         RRR      EEE
DDD         DDD      CCC                  KKK      KKK         BBB         BBB      RRR         RRR      EEE
DDD         DDD      CCC                  KKKKKKKKK            BBBBBBBBBBBB         RRRRRRRRRRRR         EEEEEEEEEEEE
DDD         DDD      CCC                  KKKKKKKKK            BBBBBBBBBBBB         RRRRRRRRRRRR         EEEEEEEEEEEE
DDD         DDD      CCC                  KKKKKKKKK            BBBBBBBBBBBB         RRRRRRRRRRRR         EEEEEEEEEEEE
DDD         DDD      CCC                  KKK      KKK         BBB         BBB      RRR   RRR            EEE
DDD         DDD      CCC                  KKK      KKK         BBB         BBB      RRR   RRR            EEE
DDD         DDD      CCC                  KKK      KKK         BBB         BBB      RRR   RRR            EEE
DDD         DDD      CCC                  KKK         KKK      BBB         BBB      RRR      RRR         EEE
DDD         DDD      CCC                  KKK         KKK      BBB         BBB      RRR      RRR         EEE
DDD         DDD      CCC                  KKK         KKK      BBB         BBB      RRR      RRR         EEE
DDDDDDDDDDDD            CCCCCCCCCCCC      KKK         KKK      BBBBBBBBBBBB         RRR         RRR      EEEEEEEEEEEEEEE
DDDDDDDDDDDD            CCCCCCCCCCCC      KKK         KKK      BBBBBBBBBBBB         RRR         RRR      EEEEEEEEEEEEEEE
DDDDDDDDDDDD            CCCCCCCCCCCC      KKK         KKK      BBBBBBBBBBBB         RRR         RRR      EEEEEEEEEEEEEEE


   SSSSSSSSSSSS      EEEEEEEEEEEEEEE         QQQQQQQQQ
   SSSSSSSSSSSS      EEEEEEEEEEEEEEE         QQQQQQQQQ
   SSSSSSSSSSSS      EEEEEEEEEEEEEEE         QQQQQQQQQ
SSS                  EEE                  QQQ         QQQ
SSS                  EEE                  QQQ         QQQ
SSS                  EEE                  QQQ         QQQ
SSS                  EEE                  QQQ         QQQ
SSS                  EEE                  QQQ         QQQ
SSS                  EEE                  QQQ         QQQ
   SSSSSSSSS         EEEEEEEEEEEE         QQQ         QQQ
   SSSSSSSSS         EEEEEEEEEEEE         QQQ         QQQ
   SSSSSSSSS         EEEEEEEEEEEE         QQQ         QQQ
            SSS      EEE                  QQQ   QQQ   QQQ
            SSS      EEE                  QQQ   QQQ   QQQ
            SSS      EEE                  QQQ   QQQ   QQQ
            SSS      EEE                  QQQ      QQQ
            SSS      EEE                  QQQ      QQQ
            SSS      EEE                  QQQ      QQQ
SSSSSSSSSSSS         EEEEEEEEEEEEEEE         QQQQQQ   QQQ
SSSSSSSSSSSS         EEEEEEEEEEEEEEE         QQQQQQ   QQQ
SSSSSSSSSSSS         EEEEEEEEEEEEEEE         QQQQQQ   QQQ
```

```
LPTSPL Version 6(100344)  Running on MTA1
 *START* User DAVIES,TOM [400,2704] Job  DCKBRE Seq. 5 Date 13-Oct-76 14:57:05 Monitor IPC-D 507B [1A3] *START*
Request created: 13-Oct-76  14:48:15      /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

[illegible]14:57:08
File will be RENAMEd to <057> protection
```

DO1



LPTSPL Version 6(100344) Running on MTA1
 START User DAVIES,TOM [400,2704] Job DCKBRE Seq. 5 Date 13-Oct-76 14:57:05 Monitor IPC-D 507B [1A3] *START*
Request created: 13-Oct-76 14:48:15   /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

[illegible]

File will be RENAMEd to <057> protection

```
.REM %
```


IDENTIFICATION

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

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

DATE CREATED: AUGUST, 1973

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

187 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 (B)

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 SW$<9,10,13,15> 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 PC	REGISTER UNDER TEST	EXPECTED ADDR.BITS	ACTUAL ADDR.BITS	EXPECTED ABORT PC	ACTUAL 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 INCORRECT
                            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 **

```
461                                     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:

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

LOCATION 1342 UP TO 1364 WILL CONTAIN ANYMORE REGISTERS FOUND

$TMPO(LOCATION 1366) WILL CONTAIN A PARITY MEMORY LOCATION CORRESPONDING TO THE REGISTER IN $REGO

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

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

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

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

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.

```
572                                     THIS VIRTUAL ADDRESS AND ITS' CORRESPONDING OFFSET VALUE
573                                     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	23700	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 ABORT 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 SOB INSTRUCTION
(C) A 'MOV SMD,DMD' 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<06>
                                        ;
                                        ;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          ERRVEC= 4
                000010          RESVEC= 10
                000014          TBITVEC=14
                000014          TRTVEC= 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
```

```
856  000052  040000                      BIT14
857
858          000204                      .=204
859
860                                      ;KT11-D STATUS REGISTER ADDRESSES
861
862  000204  177572                      SR0:    177572
863  000206  177576                      SR2:    177576
864
865                                      ;KERNAL PAGE DESCRIPTOR REGISTERS
866
867  000210  172300                      KPDR0:  172300
868  000212  172302                      KPDR1:  172302
869  000214  172304                      KPDR2:  172304
870  000216  172306                      KPDR3:  172306
871  000220  172310                      KPDR4:  172310
872  000222  172312                      KPDR5:  172312
873  000224  172314                      KPDR6:  172314
874  000226  172316                      KPDR7:  172316
875
876                                      ;KERNAL PAGE ADDRESS REGISTERS
877
878  000230  172340                      KPAR0:  172340
879  000232  172342                      KPAR1:  172342
880  000234  172344                      KPAR2:  172344
881  000236  172346                      KPAR3:  172346
882  000240  172350                      KPAR4:  172350
883  000242  172352                      KPAR5:  172352
884  000244  172354                      KPAR6:  172354
885  000246  172356                      KPAR7:  172356
886
887                                      ;KT11 VECTOR ADDRESS
888
889  000250  000250  000252              SEGVEC: 250,252
```

```
                                ;*****************************************************************************
001100                                  .=1100

                                ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.

                                ;CALL:
                                ;1) USING A TRAP INSTRUCTION
                                ;       TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;OR
                                ;       TYPE
                                ;       MESADR

                                ;2) USING A JSR INSTRUCTION
                                ;       MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
                                ;       JSR     PC,$TYPE                ;CALL TYPE ROUTINE
                                ;       MESADDR                         ;FIRST ADRESS OF MESSAGE

001100  177564          $TPS:  177564                                  ;TTY PRINTER STATUS REG. ADDRESS
001102  177566          $TPB:  177566                                  ;TTY PRINTER BUFFER REG. ADDRESS
001104     000          $NULL: .BYTE   0                               ;CONTAINS NULL CHARACTER FOR FILLS
001105     002          $FILLS: .BYTE  2                               ;CONTAINS # OF FILLER CHARACTERS REQUIRED
001106     000          $TPFLG: .BYTE  0                               ;"TERMINAL AVAILABLE" FLAG (0=YES)
001107     000                  .BYTE  0                               ;RESERVED

001110  105767  177772  $TYPE: TSTB    $TPFLG                          ;IS THERE A TERMINAL?
001114  001402                 BEQ     6$                              ;BR IF YES
001116  000000                 HALT                                    ;HALT HERE IF NO TERMINAL
001120  000407                 BR      7$                              ;LEAVE
001122  010046          6$:    MOV     R0,-(SP)                        ;SAVE R0
001124  017600  000002         MOV     @2(SP),R0                       ;GET ADDRESS OF ASCIZ STRING
001130  112046          1$:    MOVB    (R0)+,-(SP)                     ;PUSH CHARACTER TO BE TYPED ONTO STACK
001132  001005                 BNE     2$                              ;BR IF IT ISN'T THE TERMINATOR
001134  005726                 TST     (SP)+                           ;IF TERMINATOR POP IT OFF THE STACK
001136  012600                 MOV     (SP)+,R0                        ;RESTORE R0
001140  062716  000002   7$:    ADD     #2,(SP)                         ;ADJUST RETURN PC
001144  000002                 RTI                                     ;RETURN
001146  004767  000026   2$:    JSR     PC,5$                           ;GO TYPE THIS CHARACTER
001152  122726  000012   3$:    CMPB    #12,(SP)+                       ;CHECK IF THE CHAR. TYPED WAS A  LINE FEED
001156  001364                 BNE     1$                              ;GO GET NEXT CHAR. IF NOT LINE FEED
001160  016746  177720         MOV     $NULL,-(SP)                     ;GET # OF FILLER CHARS. NEEDED
                                                                        ;AND THE NULL CHAR.
001164  105366  000001   4$:    DECB    1(SP)                           ;DOES A NULL NEED TO BE TYPED?
001170  002770                 BLT     3$                              ;BR IF NO--GO POP THE NULL OFF OF STACK
001172  004767  000002         JSR     PC,5$                           ;GO TYPE A NULL
001176  000772                 BR      4$                              ;LOOP
001200  105777  177674   5$:    TSTB    @$TPS                           ;WAIT UNTIL PRINTER IS READY
001204  100375                 BPL     5$
001206  116677  000002  177666 MOVB    2(SP),@$TPB                     ;LOAD CHAR TO BE TYPED INTO DATA REG.
001214  000207                 RTS     PC
001216  000062                 .BLKB   62                              ;RESERVE SOME MORE CORE FOR OVERLAY CAPIBILITIES
```

```
                                ;*****************************************************************************
                                ;COMMON TAGS
                                        .=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
```

```
 1000  001436  000000                  $SET7:   .WORD   0
 1001  001440  000000                  $SET10:  .WORD   0
 1002  001442  000000                  $SET11:  .WORD   0
 1003  001444  000000                  $SET12:  .WORD   0
 1004  001446  000000                  NTERAD:  .WORD   0
 1005  001450  000000                  NTER0:   .WORD   0          ;THESE LOCATIONS CONTAIN THE
 1006  001452  000000                  NTER1:   .WORD   0          ;APPROPRIATE INTERLEAVE FACTORS
 1007  001454  000000                  NTER2:   .WORD   0          ;FOR THE PARITY CONTROL
 1008  001456  000000                  NTER3:   .WORD   0          ;REGISTERS (IF ANY)
 1009  001460  000000                  NTER4:   .WORD   0
 1010  001462  000000                  NTER5:   .WORD   0
 1011  001464  000000                  NTER6:   .WORD   0
 1012  001466  000000                  NTER7:   .WORD   0
 1013  001470  000000                  NTER10:  .WORD   0
 1014  001472  000000                  NTER11:  .WORD   0
 1015  001474  000000                  NTER12:  .WORD   0
 1016  001476  000000                  NEWSTK:  .WORD   0
 1017                                  ;END OF USER DEFINED TAG(S)
 1018
```

```
                                ;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
```

```
1075  001562  014540                    DH2             ;REGISTER UNDER TEST
1076  001564  015114                    DT2             ;PARITY
1077  001566  000000                    0
1078                              ;ITEM 10
1079  001570  014252                    EM10            ;DIDN'T ABORT OR RECOGNIZE
1080                                                     ;STACK VIOLATION
1081  001572  015005                    DH5             ;PROGRAM PC
1082                                                     ;REGISTER UNDER TEST
1083                                                     ;EXPECTED ABORT PC
1084  001574  015142                    DT5             ;$HLTAD, PARITY, $GDDAT
1085  001576  000000                    0
1086                              ;ITEM 11
1087  001600  014326                    EM11            ;ABORTED BUT STACK VIOLATION
1088                                                     ;NOT RECOGNIZED
1089  001602  014470                    DH1             ;PROGRAM PC
1090                                                     ;REGISTER UNDER TEST
1091  001604  015106                    DT1             ;$HLTAD, PARITY
1092  001606  000000                    0
1093                              ;ITEM 12
1094  001610  014403                    EM12            ;STACK VIOLATION PICKED UP BUT
1095                                                     ;ABORT NOT RECOGNIZED
1096  001612  014470                    DH1             ;PROGRAM PC
1097                                                     ;REGISTER UNDER TEST
1098  001614  015106                    DT1             ;$HLTAD, PARITY
1099  001616  000000                    0
```

```
1100
1101                    ;*****************************************************************
1102                    ;
1103                    ;SYSTEM PARITY REGISTER NOTES FOR MF11 AND MS11
1104                    ;
1105                    ;*****************************************************************
1106
1107                    ;BIT ASSIGNMENTS FOR THE MF11 PARITY REGISTER IS AS FOLLOWS:
1108                    ;
1109                    ;BIT15          PARITY ERROR
1110                    ;BITS 11-5      ERROR ADDRESS           ;HIGH ORDER ADDRESS BITS
1111                                                            ;OF ADDRESS OF MOST RECENT ERROR
1112                                                            ;(BITS 17 THRU 11)
1113                    ;BIT02          WRITE                   ;NORMAL PARITY (ODD) WHEN CLEAR
1114                                                            ;OTHER PARITY (EVEN) WHEN SET
1115                    ;BIT00          ERROR ACTION ENABLE     ;NO ACTION WHEN CLEAR
1116                                                            ;TRAP TO VECTOR 114 WHEN SET
1117                    ;
1118                    ;NOTE: THE ABOVE BITS ARE READ/WRITE AND CAN BE CLEARED BY 'INIT' (EXCEPT BITS 11-5)
1119                    ;
1120                    ;\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
1121                    ;
1122                    ;BIT ASSIGNMENTS FOR THE MS11 PARITY REGISTER IS AS FOLLOWS:
1123                    ;
1124                    ;BIT15          PARITY ERROR
1125                    ;BIT02          WRITE                   ;NORMAL PARITY (EVEN) WHEN CLEAR
1126                                                            ;OTHER PARITY (ODD) WHEN SET
1127                    ;BIT00          ERROR ACTION ENABLE     ;NO ACTION WHEN CLEAR
1128                                                            ;TRAP TO VECTOR 114 WHEN SET
1129                    ;
1130                    ;NOTE:  THERE ARE NO ERROR ADDRESS BITS IN THE CURRENT MS11 PARITY REGISTER
1131                    ;       HOWEVER, THERE WILL BE IN A LATER VERSION WHICH WILL BE
1132                    ;       HANDLED PROPERLY BY THIS PROGRAM
1133
1134                    ;*****************************************************************
1135                    ;*****************************************************************
1136                    ;
1137                    ;SPECIAL NOTE----THERE ARE 2 GENERAL PURPOSE REGISTERS USED IN THE
1138                    ;                 PROGRAM FOR SPECIFIC CIRCUMSTANCES.  THEY ARE:
1139                    ;
1140                    ;               R1 - WILL ALWAYS CONTAIN THE 1ST ADDRESS OF THE 2
1141                    ;                    LOCATION MAP USED FOR TESTING.
1142                    ;                    THE CONTENTS OF R1 IS DETERMINED BY THE 'COMPUT'
1143                    ;                    ROUTINE SHOWN FURTHER DOWN.
1144                    ;                    EXAMINATION OF R1 WILL TELL YOU WHERE IN PARITY
1145                    ;                    MEMORY TESTING IS BEING CONDUCTED.
1146                    ;
1147                    ;                R5 - WILL ALWAYS CONTAIN THE ADDRESS OF THE ROUTINE
1148                    ;                    FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS.
1149                    ;
1150                    ;*****************************************************************
1151                    ;*****************************************************************
```

```
1152
1153
1154
1155                    ;*****************************************************************
1156                    ;
1157                    ;MISCELLANEOUS COMMON PARITY VARIABLES AND FLAGS
1158                    ;
1159                    ;*****************************************************************
1160  001620  000114    INTVEC: 114                     ;PARITY INTERRUPT VECTOR ADDRESS
1161  001622  000000    PARITY: 0                       ;CONTAINS PARITY REGISTER IN USE
1162  001624  000000    PSPCORZONES:    0               ;FLAG TO INDICATE TO 'CHECKLOC'
1163                                                    ;ROUTINE THAT A PS OR PC FETCH
1164                                                    ;OR A ZONE ABORT WAS DONE
1165                                                    ;0 = NO;  1 = YES
1166  001626  000000    MSREGFLAG:      0               ;INDICATES PARITY TYPE
1167                                                    ;0 = CORE ; 1 = MOS
1168  001630  000000    USERTYPE:       0               ;INDICATES USER SELECTION OF
1169                                                    ;PARITY REGISTER
1170                                                    ;0 = PROGRAM FIND
1171                                                    ;1 = USER SELECTION
1172  001632  000000    BLKCNT: 0                       ;CONTAINS THE NUMBER OF CONSEC-
1173                                                    ;UTIVE LOCATIONS TO BE TESTED
1174                                                    ;DURING PROGRAM TABULATION TO
1175                                                    ;COVER CASES OF MEMORY INTER-
1176                                                    ;LEAVING
1177  001634  000000    RESTOREBASE:    0               ;HOLDS PAGE 1 ADDRESS
1178                                                    ;OR CURRENT MEMORY ADDRESS FOR
1179                                                    ;RESTORATION DURING
1180                                                    ;RUNNING OF PROGRAM.  IT IS USED IF
1181                                                    ;WE HAVE CHECKED CONSECUTIVE
1182                                                    ;LOCATIONS WITHOUT AN ABORT BEFORE
1183                                                    ;GOING TO NEXT OFFSET WHICH WILL
1184                                                    ;PUT US IN ANOTHER BANK
1185  001636  000000    LEAFCNT:        0               ;CONTAINS THE NO. OF ABORTS
1186                                                    ;ENCOUNTERED IN DETERMINATION OF
1187                                                    ;AN INTERLEAVE FACTOR
1188  001640  000000    MEMAD:  0                       ;CONTAINS A BASE ADDRESS OR A
1189                                                    ;CURRENT MEMORY ADDRESS USED IN
1190  001642  000000    CPU4O:  0                       ;FLAG TO INDICATE PROCESSOR
1191                                                    ;0 = 11/45; 1 = 11/40
1192                                                    ;PARITY TABLE CREATION
1193
1194                    ;THE FOLLOWING TABLE IS USED TO DETERMINE THE
1195                    ;INTERLEAVE FACTOR FOR THE CONTROL REGISTERS
1196  001644            INTERTABLE:
1197  001644  000003  000001   3,1                     ;3 ABORTS ON 3 CONSECUTIVE LOCS. = 1 WAY LEAVE
1198  001650  000006  000002   6,2                     ;3 ABORTS ON 6 CONSECUTIVE LOCS. = 2 WAY LEAVE
1199  001654  000011  000003   9.,3                    ;3 ABORTS ON 9 CONSECUTIVE LOCS. = 3 WAY LEAVE
1200  001660  000014  000004   12.,4                   ;3 ABORTS ON 12 CONSECUTIVE LOCS. = 4 WAY LEAVE
1201  001664  000017  000005   15.,5                   ;3 ABORTS ON 15 CONSECUTIVE LOCS. = 5 WAY LEAVE
1202  001670  000022  000006   18.,6                   ;3 ABORTS ON 18 CONSECUTIVE LOCS. = 6 WAY LEAVE
1203  001674  000025  000007   21.,7                   ;3 ABORTS ON 21 CONSECUTIVE LOCS. = 7 WAY LEAVE
1204  001700  000030  000010   24.,8.                  ;3 ABORTS ON 24 CONSECUTIVE LOCS. = 8 WAY LEAVE
1205  001704  000000           0                       ;END OF TABLE TERMINATOR
```

```
1206
1207  001706                                  BEGIN:
1208  001706  012706  001100                        MOV     #STACK,SP               ;SETUP THE STACK POINTER
1209  001712  012737  012062  000020                MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
1210  001720  012737  000340  000022                MOV     #340,@#IOTVEC+2         ;LEVEL 7
1211  001726  005067  177350                       CLR     $TSTNM                  ;INITIALIZE THE TEST NUMBER
1212  001732  012737  012634  000030                MOV     #$HLT,@#EMTVEC          ;EMT VECTOR FOR HLT(ERROR) ROUTINE
1213  001740  012737  000340  000032                MOV     #340,@#EMTVEC+2         ;LEVEL 7
1214  001746  012737  013556  000034                MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
1215  001754  012737  000340  000036                MOV     #340,@#TRAPVEC+2        ;LEVEL 7
1216  001762  012737  013606  000024                MOV     #$PWRDN,@#PWRVEC        ;POWER FAILURE VECTOR
1217  001770  012737  000340  000026                MOV     #340,@#PWRVEC+2         ;LEVEL 7
1218  001776  005067  177276                       CLR     $PASS                   ;CLEAR THE PASS COUNT
1219  002002  005067  177276                       CLR     $ICNT                   ;INITIALIZE THE ITERATION COUNTER
1220  002006  005067  010276                       CLR     $TIMES                  ;INITIALIZE NUMBER OF ITERATIONS
1221  002012  105067  177276                       CLRB    $ERFLG                  ;CLEAR THE ERROR FLAG
1222  002016  005067  177270                       CLR     $ERTTL                  ;CLEAR THE ERROR COUNT
1223  002022  005067  010746                       CLR     $ESCAPE                 ;CLEAR THE ESCAPE ON ERROR ADDRESS
1224
1225  002026  005037  001630                       CLR     @#USERTYPE              ;SET USER SELECTION INDICATOR
1226                                                                                ;TO ZERO INDICATING PROGRAM
1227                                                                                ;TABULATION
1228  002032  005037  002304                       CLR     @#$KT11                 ;CLEAR KT11 PRESENCE FLAG
1229  002036  005037  001420                       CLR     @#$SET0                 ;CLEAR THE OFFSET
1230  002042  005037  001422                       CLR     @#$SET1                 ;TABLE LOCATIONS FOR
1231  002046  005037  001424                       CLR     @#$SET2                 ;THE KT11 OPTION
1232  002052  005037  001426                       CLR     @#$SET3
1233  002056  005037  001430                       CLR     @#$SET4
1234  002062  005037  001432                       CLR     @#$SET5
1235  002066  005037  001434                       CLR     @#$SET6
1236  002072  005037  001436                       CLR     @#$SET7
1237  002076  005037  001440                       CLR     @#$SET10
1238  002102  005037  001442                       CLR     @#$SET11
1239  002106  005037  001444                       CLR     @#$SET12
1240  002112  005037  001450                       CLR     @#NTER0                 ;CLEAR THE INTERLEAVE TABLE
1241  002116  005037  001452                       CLR     @#NTER1                 ;ENTRY LOCATIONS
1242  002122  005037  001454                       CLR     @#NTER2
1243  002126  005037  001456                       CLR     @#NTER3
1244  002132  005037  001460                       CLR     @#NTER4
1245  002136  005037  001462                       CLR     @#NTER5
1246  002142  005037  001464                       CLR     @#NTER6
1247  002146  005037  001466                       CLR     @#NTER7
1248  002152  005037  001470                       CLR     @#NTER10
1249  002156  005037  001472                       CLR     @#NTER11
1250  002162  005037  001474                       CLR     @#NTER12
1251  002166  005037  001636                       CLR     @#LEAFCNT               ;CLEAR NO. OF ABORTS PER NO. OF
1252                                                                                ;CONSECUTIVE LOCS. TESTED FLAG
1253  002172  005037  001642                       CLR     @#CPU40                 ;CLEAR PROCESSOR INDICATOR FLAG
1254  002176  013746  000004                       MOV     @#4,-(SP)               ;SAVE CONTENTS OF LOC. 4
1255  002202  013746  000010                       MOV     @#10,-(SP)              ;SAVE CONTENTS OF LOC. 10
1256  002206  012737  002226  000010                MOV     #1$,@#RESVEC            ;SET UP FOR 'SPL' TRAP ADDRESS
1257  002214  012737  000340  000012                MOV     #340,@#RESVEC+2         ;SET UP FOR 'SPL' TRAP PS
1258  002222  000237                               SPL     7                       ;ATTEMPT TO SET A PRIORITY LEVEL
1259  002224  000403                               BR      2$                      ;BRANCH INDICATING WE ARE ON AN
1260                                                                                ;11/45 PROCESSOR
1261  002226  022626                        1$:    CMP     (SP)+,(SP)+             ;RESET THE STACK FROM TRAP
```

```
1262  002230  005237  001642                  INC     @#CPU40         ;SET FLAG INDICATING WE ARE ON
1263                                                                    ;AN 11/40 PROCESSOR
1264  002234  012637  000010          2$:     MOV     (SP)+,@#10      ;RESTORE CONTENTS OF LOC. 10
1265  002240  005037  000012                  CLR     @#12            ;RESTORE TRAPCATCHER LOC. 12
1266  002244  012737  002324  000004          MOV     #KTTIMEOUT,@#ERRVEC ;SET UP KT TIMEOUT ADDRESS
1267  002252  012737  000340  000006          MOV     #340,@#ERRVEC+2 ;SET UP KT TIMEOUT PS
1268  002260  005777  175720                  TST     @SR0            ;KT11 ARE YOU THERE?
1269  002264  005077  175714                  CLR     @SR0            ;YES - INITIALIZE IT IN CASE
1270                                                                    ;USER DOESN'T WANT IT
1271  002270  013700  177570                  MOV     @#SWR,R0        ;GET SWR CONTENTS
1272  002274  006300                          ASL     R0              ;MOVE BIT06 TO BIT07 POSITION
1273  002276  105700                          TSTB    R0              ;KT11 PRESENT (OBVIOUSLY) IF
1274                                                                    ;WE REACH THIS INSTRUCTION
1275                                                                    ;DOES USER WANT IT?
1276  002300  100412                          BMI     GO              ;BRANCH IF NO
1277  002302  005327                          DEC     (PC)+           ;YES - SET KT11 FLAG
1278  002304  000000          $KT11:  0                               ;CONTAINS A -1 IF KT11 OPTION
1279                                                                    ;IS PRESENT
1280  002306  004737  013124                  JSR     PC,@#$SIZE      ;SEE HOW MUCH MEMORY IS AVAILABLE
1281  002312  005077  175712                  CLR     @KPAR0          ;CLEAR PAGE 0 OFFSET REGISTER
1282  002316  005277  175662                  INC     @SR0            ;TURN ON MEMORY MANAGEMENT
1283  002322  000401                          BR      GO              ;SKIP NEXT INSTRUCTION
1284  002324  022626          KTTIMEOUT:      CMP     (SP)+,(SP)+     ;RESET THE STACK FROM TIMEOUT
1285                                                                    ;KT11 NOT PRESENT, THEREFORE
1286                                                                    ;ONLY DO BELOW 28K
1287  002326  012637  000004          GO:     MOV     (SP)+,@#4       ;RESTORE CONTENTS OF LOC. 4
1288  002332  005037  000006                  CLR     @#6             ;RESTORE CONTENTS OF LOC. 6
1289  002336  004337  011506                  JSR     R3,@#INITIALIZE ;SET UP TO BEGIN TESTING
1290  002342  016703  177046                  MOV     $TMPAD,R3       ;SET UP FOR MEMORY TABLE CREATION
1291  002346  016702  176764                  MOV     $REGAD,R2       ;SET UP FOR PARITY TABLE CREATION
1292  002352  016700  177040                  MOV     $SETAD,R0       ;SET UP FOR OFFSET TABLE CREATION
1293                                                                    ;THIS TABLE ONLY HAS EFFECT IF
1294                                                                    ;MEMORY MGMT IS TURNED ON
1295  002356  016705  177064                  MOV     NTERAD,R5       ;SET UP FOR INTERLEAVE TABLE
1296                                                                    ;CREATION (8 - WAY INTERLEAVE
1297                                                                    ;CAPABILITY EXISTS)
1298                          ;*****************************************************************
1299                          ;
1300                          ;LET'S DETERMINE IF SEVERAL REGISTERS EXIST, FOR EXAMPLE,
1301                          ;
1302                          ;       172100 GOVERNING CORE MEMORY    0 - 8K
1303                          ;   &   172102 GOVERNING MOS MEMORY     8 - 16K
1304                          ;   &   172112 GOVERNING CORE MEMORY    40 - 56K
1305                          ;
1306                          ;IF WE WANT TO PRESELECT ONE OF THEM OR CREATE A TABLE OF ALL THOSE
1307                          ;AVAILABLE AND CARRY ON TESTING FROM THE TABLE
1308                          ;
1309                          ;NOTE:  SEE DOCUMENT CONCERNING TABLE APPEARANCES AS A
1310                          ;       FUNCTION OF MEMORY MANAGEMENT (KT11 OPTION) BEING
1311                          ;       ENABLED OR DISABLED DURING PROGRAM EXECUTION
1312                          ;
1313                          ;*****************************************************************
1314  002362  032737  010000  177570          BIT     #BIT12,@#SWR    ;DOES THE USER WISH TO SELECT THE
1315                                                                    ;REGISTER?
1316  002370  001445                          BEQ     FINDONE         ;BRANCH IF NO
1317  002372          MSGTYP:
```

```
1318  002372  104400  002400                  TYPE    ,.+4                    ;TYPE ASCIZ STRING
1319  002376  000433                          BR      64$                     ;GET OVER THE ASCII
1320                                          ;.ASCIZ     <15><12>"TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN  "
1321  002466                        64$:
1322  002466  104406  001340                  ACCEPT,$REGO                    ;PICK UP THE DESIRED REGISTER
1323                                                                          ;FROM THE TELETYPE AND STORE
1324                                                                          ;IN FIRST TABLE LOCATION
1325  002472  005237  001630                  INC     @#USERTYPE              ;SET FLAG INDICATING USER SELECTION
1326  002476  013746  000004                  MOV     @#4,-(SP)               ;SAVE CONTENTS OF LOC. 4
1327  002502  000404                          BR      NEXT1                   ;SKIP THE NEXT INSTRUCTIONS
1328  002504  012712  172100        FINDONE:  MOV     #172100,(R2)            ;MOVE 1ST POSSIBLE PARITY
1329                                                                          ;REGISTER INTO $REGO
1330  002510  013746  000004                  MOV     @#4,-(SP)               ;PUSH CONTENTS OF LOC.4 ONTO STACK
1331  002514  012704  001644        NEXT1:    MOV     #INTERTABLE,R4          ;INITIALIZE INTERLEAVE TABLE
1332                                                                          ;POINTER
1333  002520  005737  002304                  TST     @#$KT11                 ;KT11 ARE YOU THERE?
1334  002524  001402                          BEQ     5$                      ;BRANCH IF NO
1335  002526  005077  175500                  CLR     @KPAR1                  ;RESET PAGE 1 ADDRESS REGISTER
1336                                                                          ;BEFORE TESTING NEXT PARITY
1337                                                                          ;CONTROL REGISTER
1338  002532  012737  003144  000004  5$:     MOV     #NOREG,@#4              ;SET PARITY TIMEOUT VECTOR SERVICE ADDRESS
1339  002540  022712  172136                  CMP     #172136,(R2)            ;IS THE ADDRESS IN BOUNDS?
1340  002544  100002                          BPL     12$                     ;BRANCH IF YES
1341  002546  000137  003172                  JMP     @#NOMORE                ;OTHERWISE - TERMINATE TABLE!
1342  002552  005772  000000        12$:     TST     @(R2)                   ;YES - IS THIS REGISTER PRESENT?
1343  002556  004737  003222                  JSR     R7,@#PARTST             ;CHECK OUT FOR FATAL ERRORS
1344                                          ;
1345                                          ;
1346                                          ;WE HAVE CHECKED OUT THE REGISTER AND FOUND IT TO BE WORKING PROPERLY
1347                                          ;NOW WE WILL FIND ITS ASSOCIATED PARITY MEMORY, IF IT EXISTS!!
1348                                          ;
1349                                          ;
1350  002562  012737  003112  000004          MOV     #PARCORE,@#4            ;SET MEMORY TIMEOUT VECTOR
1351                                                                          ;SERVICE ADDRESS
1352  002570  012737  013700  001640          MOV     #13700,@#MEMAD          ;SET UP A STARTING ADDRESS
1353  002576  011437  001632                  MOV     (R4),@#BLKCNT           ;SET A COUNTER FOR CONSECUTIVE
1354                                                                          ;LOCATION CHECKS TO COVER MEMORY
1355                                                                          ;INTERLEAVING
1356  002602  005737  002304                  TST     @#$KT11                 ;SHOULD I LOOK ABOVE 28K?
1357  002606  100013                          BPL     1$                      ;BRANCH IF NO
1358  002610  062737  010000  001640          ADD     #10000,@#MEMAD          ;STEP UP TO A PAGE 1 BASE ADDRESS
1359                                                                          ;IF MEMORY MANAGEMENT TURNED ON
1360  002616  013737  001640  001634          MOV     @#MEMAD,@#RESTOREBASE   ;SAVE PAGE 1 BASE ADDRESS
1361  002624  062710  000140          2$:     ADD     #140,(R0)               ;SET UP AN OFFSET FOR KPAR1
1362  002630  011077  175376                  MOV     (R0),@KPAR1             ;SET OFFSET IN PAGE 1 REGISTER
1363  002634  000406                          BR      9$                      ;SKIP NEXT 2 INSTRUCTIONS
1364  002636  062737  004000  001640  1$:     ADD     #4000,@#MEMAD           ;STEP UP TO NEXT BANK
1365  002644  013737  001640  001634          MOV     @#MEMAD,@#RESTOREBASE   ;SAVE INITIAL MEMORY ADDRESS
1366  002652  005777  176762          9$:     TST     @MEMAD                  ;IS THIS MEMORY AVAILABLE?
1367  002656  013713  001640                  MOV     @#MEMAD,(R3)            ;YES - STORE THIS MEMORY LOCATION
1368  002662  004737  003430                  JSR     R7,@#ABORT              ;NOW LET'S SEE IF IT'S PARITY
1369                                                                          ;MEMORY CORRESPONDING TO THE
1370                                                                          ;PARITY REGISTER WE'VE FOUND
1371  002666  005737  002304                  TST     @#$KT11                 ;KT11 ARE YOU THERE?
1372  002672  100034                          BPL     4$                      ;BRANCH IF NO
1373  002674  005337  001632                  DEC     @#BLKCNT                ;DECREASE CONSECUTIVE
```

```
1374
1375  002700  005737  001632                    TST     @#BLKCNT        ;LOCATION COUNTER
1376                                                                      ;ARE WE DONE CHECKING
1377  002704  001404                            BEQ     6$              ;CONSECUTIVE LOCATIONS?
1378  002706  062737  000002  001640            ADD     #2,@#MEMAD      ;BRANCH IF YES
1379  002714  000756                            BR      9$              ;STEP UP 1 LOCATION
1380                                                                      ;GO BACK TO TEST WITH THIS
1381  002716  013737  001634  001640  6$:       MOV     @#RESTOREBASE,@#MEMAD   ;LOCATION
1382                                                                      ;RESTORE PAGE 1 BASE ADDRESS
1383                                                                      ;BEFORE GOING BACK TO INCREASE
1384  002724  062704  000004                    ADD     #4,R4           ;OFFSET
1385                                                                      ;STEP TABLE POINTER UP FOR
1386                                                                      ;NEXT VALUE OF CONSECUTIVE
1387  002730  005714                            TST     (R4)            ;LOCATION TO BE CHECKED
1388  002732  001405                            BEQ     10$             ;ARE THERE ANY MORE?
1389  002734  011437  001632                    MOV     (R4),@#BLKCNT   ;BRANCH IF NO
1390                                                                      ;STORE THIS VALUE OF CONSECUTIVE
1391  002740  005037  001636                    CLR     @#LEAFCNT       ;LOCATION CHECKS
1392                                                                      ;CLEAR INTERLEAVE VALUE HOLDER
1393  002744  000742                            BR      9$              ;BEFORE RETESTING
1394                                                                      ;GO BACK TO TEST WITH THIS VALUE
1395  002746  012704  001644          10$:      MOV     #INTERTABLE,R4  ;OF CONSECUTIVE LOCATIONS
1396                                                                      ;INITIALIZE INTERLEAVE TABLE
1397  002752  011437  001632                    MOV     (R4),@#BLKCNT   ;POINTER
1398                                                                      ;RESET THE CONSECUTIVE
1399  002756  005037  001636                    CLR     @#LEAFCNT       ;LOCATION COUNTER
1400                                                                      ;CLEAR INTERLEAVE VALUE HOLDER
1401  002762  000720                            BR      2$              ;BEFORE RETESTING
1402                                                                      ;GO BACK TO INCREASE OFFSET
1403  002764  022737  157700  001640  4$:       CMP     #157700,@#MEMAD ;AND TEST
1404  002772  001434                            BEQ     8$              ;ARE WE UP TO 28K YET?
1405  002774  005337  001632                    DEC     @#BLKCNT        ;BRANCH IF YES
1406                                                                      ;DECREASE CONSECUTIVE
1407  003000  005737  001632                    TST     @#BLKCNT        ;LOCATION COUNTER
1408                                                                      ;ARE WE DONE CHECKING CONSEC-
1409  003004  001404                            BEQ     7$              ;UTIVE LOCATIONS?
1410  003006  062737  000002  001640            ADD     #2,@#MEMAD      ;BRANCH IF YES
1411  003014  000716                            BR      9$              ;STEP UP 1 LOCATION
1412                                                                      ;GO BACK TO TEST WITH THIS
1413  003016  013737  001634  001640  7$:       MOV     @#RESTOREBASE,@#MEMAD   ;LOCATION
1414                                                                      ;RESTORE INITIAL MEMORY
1415                                                                      ;ADDRESS BEFORE GOING BACK TO
1416  003024  062704  000004                    ADD     #4,R4           ;STEP UP TO NEXT BANK
1417                                                                      ;STEP TABLE POINTER UP FOR
1418                                                                      ;NEXT VALUE OF CONSECUTIVE
1419  003030  005714                            TST     (R4)            ;LOCATIONS TO BE CHECKED
1420  003032  001405                            BEQ     11$             ;ARE THERE ANY MORE?
1421  003034  011437  001632                    MOV     (R4),@#BLKCNT   ;BRANCH IF NO
1422                                                                      ;STORE THIS VALUE OF
1423  003040  005037  001636                    CLR     @#LEAFCNT       ;CONSECUTIVE LOCATION CHECKS
1424                                                                      ;CLEAR INTERLEAVE VALUE HOLDER
1425  003044  000702                            BR      9$              ;BEFORE RETESTING
1426                                                                      ;GO BACK TO STEP UP TO THE
1427  003046  012704  001644          11$:      MOV     #INTERTABLE,R4  ;NEXT BANK TO CONDUCT TESTING
1428                                                                      ;INITIALIZE INTERLEAVE
1429  003052  011437  001632                    MOV     (R4),@#BLKCNT   ;TABLE POINTER
                                                                        ;RESET THE CONSECUTIVE
```

```
1430
1431  003056  005037  001636                CLR     @#LEAFCNT       ;LOCATION COUNTER
1432                                                                  ;CLEAR INTERLEAVE VALUE HOLDER
1433  003062  000665                       BR      1$              ;BEFORE RETESTING
1434                                                                  ;GO BACK TO STEP UP TO NEXT
1435  003064  011237  001622          8$:   MOV     (R2),@#PARITY   ;BANK TO CONDUCT TESTING
1436                                                                  ;STORE THE BAD REGISTER WITH
1437  003070  104004                       HLT     +4              ;NO PARITY MEMORY
1438                                                                  ;NO PARITY MEMORY FOUND
1439  003072  005737  001630              TST     @#USERTYPE      ;BELOW 28K!!!!!!!
1440  003076  001402                       BEQ     3$              ;DID USER SELECT REGISTER?
1441  003100  000137  002372              JMP     @#MSGTYP        ;BRANCH IF NO
1442                                                                  ;GO BACK TO RETYPE MESSAGE FOR
1443  003104  062712  000002          3$:   ADD     #2,(R2)         ;USER RESPONSE
1444                                                                  ;PLACE NEXT POSSIBLE REGISTER
1445  003110  000601                       BR      NEXT1           ;INTO SAME TABLE LOCATION
1446  003112  022626                 PARCORE: CMP    (SP)+,(SP)+     ;GO BACK TO TEST THIS REGISTER
1447  003114  011237  001622              MOV     (R2),@#PARITY   ;RESET STACK FROM MEMORY TIMEOUT
1448                                                                  ;STORE THE REGISTER THAT EN-
1449                                                                  ;COUNTERED A POSSIBLE HOLE IN
1450  003120  104007                       HLT     +7              ;MEMORY
1451                                                                  ;A POSSIBLE HOLE IN MEMORY EXISTS
1452  003122  005737  001630              TST     @#USERTYPE      ;WITH NO PARITY BELOW IT!!!!!!!
1453  003126  001402                       BEQ     4$              ;DID USER SELECT REGISTER?
1454  003130  000137  002372              JMP     @#MSGTYP        ;BRANCH IF NO
1455                                                                  ;GO BACK TO RETYPE MESSAGE FOR
1456  003134  062712  000002          4$:   ADD     #2,(R2)         ;USER RESPONSE
1457                                                                  ;PLACE NEXT POSSIBLE REGISTER
1458  003140  000137  002514              JMP     @#NEXT1         ;INTO SAME TABLE LOCATION
1459  003144  022626                 NOREG: CMP     (SP)+,(SP)+     ;GO BACK TO TEST THIS REGISTER
1460  003146  005737  001630              TST     @#USERTYPE      ;RESET STACK FROM REGISTER TIMEOUT
1461  003152  001403                       BEQ     1$              ;DID THE USER SELECT THE REGISTER?
1462  003154  104006                       HLT     +6              ;BRANCH IF NO
1463                                                                  ;YES - USER SELECTED REGISTER NOT
1464  003156  000137  002372              JMP     @#MSGTYP        ;PRESENT ON SYSTEM
1465  003162  062712  000002          1$:   ADD     #2,(R2)         ;GO BACK TO RETYPE MESSAGE
1466                                                                  ;STEP UP TO NEXT PARITY REGISTER
1467  003166  000137  002514              JMP     @#NEXT1         ;AT SAME TABLE LOCATION
1468                                                                  ;PREVIOUS PARITY REGISTER NOT
1469  003172  012637  000004        NOMORE: MOV     (SP)+,@#4       ;PRESENT - SEE IF THE NEXT ONE IS
1470  003176  005012                       CLR     (R2)            ;RESTORE CONTENTS OF LOC. 4
1471                                                                  ;ALL DONE TABLE CREATION
1472  003200  000137  003620              JMP     @#START         ;END IT WITH A '0'
1473                                                                  ;START RUNNING PROGRAM WITH
1474                                                                  ;TABLE CONTENTS
1475                                 ;*****************************************************************
1476                                 ;
1477                                 ;THE FOLLOWING ROUTINE WILL CREATE A 2 LOCATION MEMORY MAP AT
1478                                 ;THE HIGH END OF A 1K CORE SECTION.  THIS 2 LOCATION MAP WILL
1479                                 ;INITIALLY BE USED TO DETERMINE WHERE/IF PARITY MEMORY
1480                                 ;PRESIDES AND LATER FOR SUBSEQUENT PROGRAM TESTING OF A REGISTER
1481                                 ;
1482  003204  162700  000376        ;*****************************************************************
1483                                 COMPUT: SUB    #376,R0         ;DROP DOWN SO AS NOT TO
1484  003210  010001                       MOV     R0,R1           ;DESTROY .ABS LOADER
1485                                                                  ;R1 CONTAINS BEGINNING ADDRESS
                                                                    ;OF MEMORY MAP
```

```
Z 1486  003212  020020                        CMP     R0,(R0)+                ;STEP R0 TO NEXT ADDRESS
  1487  003214  010011                        MOV     R0,@R1                  ;1ST MEMORY LOCATION
  1488  003216  011110                        MOV     @R1,@R0                 ;2ND MEMORY LOCATION
  1489  003220  000203                        RTS     R3                      ;RETURN TO TEST A DATI
  1490                                                                        ;WITH CONTENTS OF THESE 2 LOCS.
  1491
  1492                                ;*********************************************************************
  1493                                ;
  1494                                ;THIS ROUTINE WILL CHECK IF THE PARITY REGISTER IS STATICALLY IN
  1495                                ;GOOD OPERATION FOR TESTING TO BE CONDUCTED
  1496                                ;
  1497                                ;*********************************************************************
  1498  003222  011267  176374       PARTST: MOV     (R2),PARITY             ;GET PARITY REGISTER TO BE USED
  1499                                ;*************************************************************
  1500                                ;TEST 1 SET BIT0 (USED) OF PARITY REGISTER
  1501                                ;*************************************************************
  1502  003226  000004               TST1:   SCOPE
  1503  003230  052777  000001  176364        BIS     #BIT0,@PARITY
  1504  003236  032777  000001  176356        BIT     #BIT0,@PARITY           ;DID IT SET?
  1505  003244  001001                        BNE     .+4                     ;YES
  1506  003246  104002                        HLT     +2                      ;NO - FATAL ERROR TO PROGRAM!
  1507                                ;*************************************************************
  1508                                ;TEST 2 CLEAR BIT0 (USED) OF PARITY REGISTER
  1509                                ;*************************************************************
  1510  003250  000004               TST2:   SCOPE
  1511  003252  042777  000001  176342        BIC     #BIT0,@PARITY
  1512  003260  032777  000001  176334        BIT     #BIT0,@PARITY           ;DID IT CLEAR?
  1513  003266  001401                        BEQ     .+4                     ;YES
  1514  003270  104002                        HLT     +2                      ;NO - FATAL ERROR TO PROGRAM!
  1515                                ;*************************************************************
  1516                                ;TEST 3 SET AND CLEAR BIT2 (USED) OF PARITY REGISTER
  1517                                ;*************************************************************
  1518  003272  000004               TST3:   SCOPE
  1519  003274  052777  000004  176320        BIS     #BIT2,@PARITY
  1520  003302  032777  000004  176312        BIT     #BIT2,@PARITY           ;DID IT SET?
  1521  003310  001001                        BNE     .+4                     ;YES
  1522  003312  104002                        HLT     +2                      ;NO - FATAL ERROR TO PROGRAM!
  1523  003314  042777  000004  176300        BIC     #BIT2,@PARITY
  1524  003322  032777  000004  176272        BIT     #BIT2,@PARITY           ;DID IT CLEAR?
  1525  003330  001401                        BEQ     .+4                     ;YES
  1526  003332  104002                        HLT     +2                      ;NO - FATAL ERROR TO PROGRAM!
  1527                                ;*************************************************************
  1528                                ;TEST 4 TEST RESET ON BITS 0, 2 AND 15
  1529                                ;*************************************************************
  1530  003334  000004               TST4:   SCOPE
  1531  003336  005737  002304               TST     @#$KT11                 ;KT11 ON?
  1532  003342  100415                        BMI     WHICH1                  ;BRANCH IF YES AND DON'T DO
  1533                                                                        ;THIS TEST BECAUSE THE 'RESET'
  1534                                                                        ;WILL CLOBBER SEGMENTATION
  1535  003344  052777  100005  176250        BIS     #100005,@PARITY
  1536  003352  000005                        RESET                           ;EXPECT BITS 0, 2 AND 15 TO CLEAR
  1537  003354  032777  100005  176240        BIT     #100005,@PARITY         ;DID THEY CLEAR?
  1538  003362  001404                        BEQ     .+12                    ;YES
  1539  003364  042777  100005  176230        BIC     #100005,@PARITY         ;NO - CLEAR OUT REGISTER AS A
  1540                                                                        ;PRECAUTION
  1541  003372  104005                        HLT     +5                      ;RESET DOESN'T WORK
```

```
1542                                    ;*****************************************************************
1543                                    ;TEST 5 WHICH OPTION IS ABOUT TO BE TESTED
1544                                    ;*****************************************************************
1545  003374  000004                    TST5:   SCOPE
1546
1547  003376  052777  007740  176216    WHICH1: BIS     #7740,@PARITY           ;IS AN OLD MS11 OPTION
1548                                                                            ;WITH NO ADDRESS BITS
1549                                                                            ;ABOUT TO BE TESTED?
1550  003404  032777  007740  176210            BIT     #7740,@PARITY           ;ADDRESS BITS ABLE TO BE SET?
1551  003412  001402                            BEQ     1$                      ;BRANCH IF NO INDICATING MS11
1552  003414  000004                            SCOPE
1553  003416  000207                            RTS     R7                      ;RETURN TO NORMAL FLOW
1554  003420  005237  001626            1$:     INC     @#MSREGFLAG             ;SET FLAG INDICATING MS11 OPTION
1555                                                                            ;WITH NO ADDRESS BITS
1556  003424  000004                            SCOPE
1557  003426  000207                            RTS     R7                      ;RETURN TO NORMAL FLOW
1558
1559                                    ;*****************************************************************
1560                                    ;
1561                                    ;THE FOLLOWING ROUTINE WILL TAKE EACH 1K BANK OF MEMORY
1562                                    ;THAT IS AVAILABLE AND PERFORM A DATI IN IT
1563                                    ;TO DETERMINE IF PARITY EXISTS THERE.  THIS ROUTINE IS
1564                                    ;ONLY USED DURING TABLE CREATION
1565                                    ;
1566                                    ;*****************************************************************
1567  003430  010546                    ABORT:  MOV     R5,-(SP)                ;SAVE R5 CONTENTS ON STACK
1568  003432  010046                            MOV     R0,-(SP)                ;SAVE R0 CONTENTS ON STACK
1569  003434  011300                            MOV     (R3),R0                 ;GET THE MEMORY LOCATION
1570                                                                            ;JUST DETERMINED
1571  003436  004337  003204                    JSR     R3,@#COMPUT             ;COMPUTE AN AREA IN THIS BANK
1572                                                                            ;FOR DETERMINING PARITY MEMORY
1573  003442  011267  176154                    MOV     (R2),PARITY             ;GET THE PARITY REGISTER JUST
1574                                                                            ;FOUND AND TEST WITH IT
1575                                    ;*****************************************************************
1576                                    ;               TEST A DATI IN THIS BANK
1577                                    ;*****************************************************************
1578                                    ;                 11/45 **** ROM STATE 221 ****
1579                                    ;
1580                                    ;       11/40 **** ROM STATE 207 ****
1581  003446  012705  011450                    MOV     #VECSET,R5              ;SET UP SERVICE ROUTINE ADDRESS
1582  003452  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1583  003454  003500                            ONETRY                          ;ROUTINE ADDRESS
1584  003456  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
1585  003460  010010                            MOV     R0,@R0                  ;DO THE DATO
Z 1586  003462  010030                            MOV     R0,@(R0)+               ;DO A DATI
1587  003464  042777  000005  176130            BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
1588  003472  012600                            MOV     (SP)+,R0                ;RESTORE R0 CONTENTS
1589  003474  012605                            MOV     (SP)+,R5                ;RESTORE R5 CONTENTS
1590  003476  000207                            RTS     R7                      ;NOT PARITY MEMORY!
1591                                                                            ;RETURN TO TEST AT NEXT
1592                                                                            ;INCREMENT
1593  003500  042777  000005  176114    ONETRY: BIC     #BIT2!BIT0,@PARITY      ;WE HAVE PARITY MEMORY - PROCEED
1594  003506  016600  000004                    MOV     4(SP),R0                ;RESTORE R0 CONTENTS
1595  003512  016605  000006                    MOV     6(SP),R5                ;RESTORE R5 CONTENTS
1596  003516  005237  001636                    INC     @#LEAFCNT               ;INCREMENT INTERLEAVE COUNTER
1597  003522  022737  000003  001636            CMP     #3,@#LEAFCNT            ;3 ABORTS REACHED?
```

```
1598  003530  001403                    BEQ     1$                  ;BRANCH IF YES
1599  003532  062706  000010            ADD     #10,SP              ;BYPASS JUNK ON STACK
1600  003536  000207                    RTS     R7                  ;RETURN TO TEST AT NEXT INCREMENT
1601  003540  022626            1$:     CMP     (SP)+,(SP)+         ;POP STACK BACK FROM PARITY ABORT
1602  003542  012600                    MOV     (SP)+,R0            ;RESTORE R0 CONTENTS
1603  003544  012605                    MOV     (SP)+,R5            ;RESTORE R5 CONTENTS
1604  003546  005726                    TST     (SP)+               ;POP STACK ONCE FOR ABORT ROUTINE
1605                                                                ;ENTRY
1606  003550  005737  001630            TST     @#USERTYPE          ;DID USER TYPE IN REGISTER?
1607  003554  001404                    BEQ     2$                  ;BRANCH IF NO
1608  003556  016415  000002            MOV     2(R4),(R5)          ;SET INTERLEAVE VALUE INTO
1609                                                                ;TABLE
1610  003562  000137  003620            JMP     @#START             ;AND LOCK ON THE USER SELECTED
1611                                                                ;REGISTER FOR TESTING
1612  003566  005723            2$:     TST     (R3)+               ;USER DIDN'T SELECT - SO STEP UP
1613                                                                ;TO NEXT MEMORY TABLE LOCATION
1614  003570  005720                    TST     (R0)+               ;STEP UP TO NEXT OFFSET TABLE
1615                                                                ;LOCATION - THIS TABLE WILL ONLY
1616                                                                ;BE APPLICABLE IF MEMORY MGMT
1617                                                                ;IS TURNED ON
1618  003572  012212                    MOV     (R2)+,(R2)          ;SET NEXT POSSIBLE REGISTER INTO
1619  003574  062712  000002            ADD     #2,(R2)             ;NEXT REGISTER TABLE LOCATION
1620  003600  016425  000002            MOV     2(R4),(R5)+         ;SET INTERLEAVE VALUE INTO
1621                                                                ;TABLE
1622  003604  005037  001636            CLR     @#LEAFCNT           ;RESET NO. OF ABORTS COUNTER
1623  003610  005037  001626            CLR     @#MSREGFLAG         ;CLEAR PARITY TYPE INDICATOR
1624  003614  000137  002514            JMP     @#NEXT1             ;GO BACK TO CHECK NEXT POSSIBLE
1625                                                                ;PARITY REGISTER
1626
1627
1628
1629  ;*********************************************************************
1630  ;*********************************************************************
1631  ;*********************************************************************
1632  ;*********************************************************************
1633
1634
1635  ;IF WE HAVE REACHED THIS POINT IN THE PROGRAM THEN,
1636
1637  ;$REG0(LOCATION 1340) WILL CONTAIN THE FIRST PARITY REGISTER
1638
1639  ;                    LOCATION 1342 UP TO 1364 WILL CONTAIN ANY MORE
1640  ;                    REGISTERS FOUND
1641  ;$TMP0(LOCATION 1366) WILL CONTAIN AN ADDRESS IN THE BANK
1642  ;                         THAT HAS PARITY ASSOCIATED WITH THE PARITY
1643  ;                         REGISTER IN $REG0
1644
1645  ;                    LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING
1646  ;                    MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS
1647  ;$SET0(LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED
1648  ;                         WITH THE CORRESPONDING VALUE IN $TMP0
1649
1650  ;                    LOCATION 1422 TO 1444 WILL CONTAIN THE CORRESPONDING
1651  ;                    OFFSET VALUES FOR THE OTHER REGISTERS
1652
1653  ;NTER0(LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED
```

```
1654                                    ;                   WITH THE PARITY REGISTER IN $REGO
1655                                    ;
1656                                    ;               LOCATION 1452 TO 1474 WILL CONTAIN THE CORRESPONDING
1657                                    ;                 INTERLEAVE FACTORS FOR THE OTHER REGISTERS
1658                                    ;
1659                                    ;NOTE:  TO UNDERSTAND THE TABLE SETUP SEE THE DOCUMENT
1660                                    ;
1661                                    ;*****************************************************************************
1662                                    ;*****************************************************************************
1663                                    ;*****************************************************************************
1664                                    ;*****************************************************************************
1665
1666
1667                                    ;*****************************************************************************
1668                                    ;
1669                                    ;THE REST OF THE PROGRAM WILL TEST ALL PARITY ABORTS
1670                                    ;USING EITHER THE PARITY REGISTER TYPED BY THE USER OR THOSE
1671                                    ;ENCOUNTERED IN THE TABLE PREVIOUSLY GENERATED.
1672                                    ;
1673                                    ;*****************************************************************************
1674
1675  003620  005777  175512        START:  TST     @$REGAD                 ;ARE WE AT END OF TABLE?
1676  003624  001401                        BEQ     1$                      ;BRANCH IF YES
1677  003626  000402                        BR      2$                      ;OTHERWISE SKIP NEXT INSTRUCTION
1678  003630  000137  011464        1$:     JMP     @#RESTART               ;SET UP TO START TABLE OVER!!
1679  003634  017767  175476  175760 2$:     MOV     @$REGAD,PARITY          ;GET PARITY REGISTER TO BE USED
1680  003642  162767  000002  175432        SUB     #2,$TSTNM               ;DECREASE TEST NO. FOR SCOPE
1681                                                                        ;LOOPING IN CASE TEST #4 WAS
1682                                                                        ;DONE BECAUSE TEST #4 BEING
1683                                                                        ;DONE THROWS $TSTNM COUNT OFF
1684                                                                        ;BY 2
1685  003650  005737  002304                TST     @#$KT11                 ;KT11 ARE YOU THERE?
1686  003654  100006                        BPL     3$                      ;BRANCH IF NO
1687  003656  017777  175534  174346        MOV     @$SETAD,@KPAR1          ;GET THE OFFSET NEEDED
1688  003664  062767  000001  175410        ADD     #1,$TSTNM               ;SET TEST NO. TO A CORRECT
1689                                                                        ;VALUE; TEST #4 WAS NOT
1690                                                                        ;EXECUTED
1691  003672  017700  175516        3$:     MOV     @$TMPAD,R0              ;GET PARITY MEMORY ASSOCIATED
1692                                                                        ;WITH THE PARITY REGISTER
1693  003676  004337  003204                JSR     R3,@#COMPUT             ;COMPUTE AN AREA IN THIS BANK
1694                                                                        ;FOR TESTING
1695  003702  010137  001476                MOV     R1,@#NEWSTK             ;SET UP A NEW STACK POINTER
1696                                                                        ;FOR STACK OPERATIONS IN CASE
1697                                                                        ;NO PARITY MEMORY RESIDES IN
1698                                                                        ;LOWER 4K
1699  003706  004737  003376                JSR     PC,@#WHICH1             ;DETERMINE IF WE ARE ABOUT TO
1700                                                                        ;TEST AN OLD MOS DESIGN!
1701  003712  012705  011450                MOV     #VECSET,R5              ;SET UP SERVICE ROUTINE ADDRESS
1702                                    ;*****************************************************************
1703                                    ;TEST 6 TEST (ADDRESS)  SM0,DM3  MOV INSTRUCTION
1704                                    ;*****************************************************************
1705  003716  000004                TST6:   SCOPE
1706                                    ;                   11/45 **** ROM STATE 221 ****
1707                                    ;
1708                                    ;       11/40 **** ROM STATE 207 ****
1709  003720  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
```

```
                1710  003722  003752                          A
                1711  003724  011100                          MOV     @R1,R0          ;SET UP FOR DATO
                1712  003726  010010                          MOV     R0,@R0          ;DO THE DATO
                1713  003730  012737  003740  001332          MOV     #.+10,@#$GDDAT  ;STORE THE PC THAT SHOULD
                1714                                                                  ;BE PUSHED ON THE STACK
                1715                                                                  ;IF A PARITY ABORT OCCURS
Z               1716  003736  010030                          MOV     R0,@(R0)+       ;DO A DATI
                1717  003740  042777  000004  175654          BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
                1718  003746  104001                          HLT     +1              ;DIDN'T ABORT
                1719  003750  000410                          BR      .+22            ;GO TO NEXT TEST
                1720  003752  042777  000005  175642  A:      BIC     #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
                1721  003760  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
                1722  003764  104003                          HLT     +3              ;ABORTED INCORRECTLY
                1723  003766  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK
                1724                          ;*****************************************************************
                1725                          ;TEST 7 TEST (ADDRESS)  SM0,DM5  MOV INSTRUCTION
                1726                          ;*****************************************************************
                1727  003772  000004          TST7:   SCOPE
                1728                          ;                   11/45 **** ROM STATE 231 ****
                1729                          ;
                1730                          ;       11/40 **** ROM STATE 207 ****
                1731  003774  004015                          JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
                1732  003776  004026                          A0                      ;ROUTINE ADDRESS
                1733  004000  011100                          MOV     @R1,R0          ;SET UP FOR DATO
Z               1734  004002  010020                          MOV     R0,(R0)+        ;DO THE DATO
                1735  004004  012737  004014  001332          MOV     #.+10,@#$GDDAT  ;STORE THE PC THAT SHOULD
                1736                                                                  ;BE PUSHED ON THE STACK
                1737                                                                  ;IF A PARITY ABORT OCCURS
Z               1738  004012  010050                          MOV     R0,@-(R0)       ;DO A DATI
                1739  004014  042777  000004  175600          BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
                1740  004022  104001                          HLT     +1              ;DIDN'T ABORT
                1741  004024  000410                          BR      .+22            ;GO TO NEXT TEST
                1742  004026  042777  000005  175566  A0:     BIC     #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
                1743  004034  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
                1744  004040  104003                          HLT     +3              ;ABORTED INCORRECTLY
                1745  004042  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK
                1746                          ;*****************************************************************
                1747                          ;TEST 10        TEST (DATA)  SM1,DM6  MOV INSTRUCTION
                1748                          ;*****************************************************************
                1749  004046  000004          TST10:  SCOPE
                1750                          ;                   11/45 **** ROM STATE 27 ****
                1751                          ;
                1752                          ;       11/40 **** ROM STATE 206 ****
                1753  004050  004015                          JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
                1754  004052  004104                          A1                      ;ROUTINE ADDRESS
                1755  004054  011100                          MOV     @R1,R0          ;SET UP FOR DATO
                1756  004056  010010                          MOV     R0,@R0          ;DO THE DATO
                1757  004060  012737  004070  001332          MOV     #.+10,@#$GDDAT  ;STORE THE PC THAT SHOULD
                1758                                                                  ;BE PUSHED ON THE STACK
                1759                                                                  ;IF A PARITY ABORT OCCURS
                1760  004066  011060  177776                  MOV     @R0,-2(R0)      ;DO A DATI
                1761  004072  042777  000004  175522          BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
                1762  004100  104001                          HLT     +1              ;DIDN'T ABORT
                1763  004102  000410                          BR      .+22            ;GO TO NEXT TEST
                1764  004104  042777  000005  175510  A1:     BIC     #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
                1765  004112  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
```

```
                                        HLT     +3                      ;ABORTED INCORRECTLY
                                        MOV     #STACK,SP               ;RESET THE STACK
                                ;***************************************************************
                                ;TEST 11        TEST (ADDRESS)  SMO,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,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004144  010070  177776                  MOV     R0,@-2(R0)              ;DO A DATI
004150  042777  000004  175444          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004156  104001                          HLT     +1                      ;DIDN'T ABORT
004160  000410                          BR      .+22                    ;GO TO NEXT TEST
004162  042777  000005  175432  A2:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004170  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004174  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004176  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;***************************************************************
                                ;TEST 12        TEST (DATA)  SMO,DM2  CMP INSTRUCTION
                                ;***************************************************************
004202  000004                  TST12:  SCOPE
                                ;                       11/45 **** ROM STATE 175 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004204  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004206  004236                          B                               ;ROUTINE ADDRESS
004210  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004212  010010                          MOV     R0,@R0                  ;DO THE DATO
004214  012737  004224  001332          MOV     #.+10,@#$GDDAT          ;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  B:     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)  SMO,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                          BO                              ;ROUTINE ADDRESS
004264  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
```

```
Z 1822  004266  010020                      MOV     R0,(R0)+                ;DO THE DATO
  1823  004270  012737  004300  001332      MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
  1824                                                                        ;BE PUSHED ON THE STACK
  1825                                                                        ;IF A PARITY ABORT OCCURS
Z 1826  004276  020040                      CMP     R0,-(R0)                ;DO A DATIP, DATI
  1827  004300  042777  000004  175314      BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
  1828  004306  104001                      HLT     +1                      ;DIDN'T ABORT
  1829  004310  000410                      BR      .+22                    ;GO TO NEXT TEST
  1830  004312  042777  000005  175302  B0: BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
  1831  004320  004037  011550              JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT,
  1832  004324  104003                      HLT     +3                      ;ABORTED INCORRECTLY
  1833  004326  012706  001100              MOV     #STACK,SP               ;RESET THE STACK
  1834                          ;*********************************************************************
  1835                          ;TEST 14        TEST (DATA)  SM0,DM6  CMP INSTRUCTION
  1836                          ;*********************************************************************
  1837  004332  000004         TST14:  SCOPE
  1838                          ;                       11/45 **** ROM STATE 177 ****
  1839                          ;
  1840                          ;       11/40 **** ROM STATE 267 ****
  1841  004334  004015                      JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
  1842  004336  004370                      B1                              ;ROUTINE ADDRESS
  1843  004340  011100                      MOV     @R1,R0                  ;SET UP FOR DATO
Z 1844  004342  010020                      MOV     R0,(R0)+                ;DO THE DATO
  1845  004344  012737  004356  001332      MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
  1846                                                                        ;BE PUSHED ON THE STACK
  1847                                                                        ;IF A PARITY ABORT OCCURS
  1848  004352  020060  177776              CMP     R0,-2(R0)               ;DO A DATI, DATIP
  1849  004356  042777  000004  175236      BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
  1850  004364  104001                      HLT     +1                      ;DIDN'T ABORT
  1851  004366  000410                      BR      .+22                    ;GO TO NEXT TEST
  1852  004370  042777  000005  175224  B1: BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
  1853  004376  004037  011550              JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
  1854  004402  104003                      HLT     +3                      ;ABORTED INCORRECTLY
  1855  004404  012706  001100              MOV     #STACK,SP               ;RESET THE STACK
  1856                          ;*********************************************************************
  1857                          ;TEST 15        TEST (DATA)  SM0,DM1  CMP INSTRUCTION
  1858                          ;*********************************************************************
  1859  004410  000004         TST15:  SCOPE
  1860                          ;                       11/45 **** ROM STATE 175 ****
  1861                          ;
  1862                          ;       11/40 **** ROM STATE 267 ****
  1863  004412  004015                      JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
  1864  004414  004444                      B2                              ;ROUTINE ADDRESS
  1865  004416  011100                      MOV     @R1,R0                  ;SET UP FOR DATO
  1866  004420  010010                      MOV     R0,@R0                  ;DO THE DATO
  1867  004422  012737  004432  001332      MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
  1868                                                                        ;BE PUSHED ON THE STACK
  1869                                                                        ;IF A PARITY ABORT OCCURS
  1870  004430  020010                      CMP     R0,@R0                  ;DO A DATIP, DATI
  1871  004432  042777  000004  175162      BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
  1872  004440  104001                      HLT     +1                      ;DIDN'T ABORT
  1873  004442  000410                      BR      .+22                    ;GO TO NEXT TEST
  1874  004444  042777  000005  175150  B2: BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
  1875  004452  004037  011550              JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
  1876  004456  104003                      HLT     +3                      ;ABORTED INCORRECTLY
  1877  004460  012706  001100              MOV     #STACK,SP               ;RESET THE STACK
```

```
                                ;*****************************************************************
                                ;TEST 16        TEST (DATA)  SMO,DM3  CMP INSTRUCTION
                                ;*****************************************************************
004464  000004                  TST16:  SCOPE
                                ;                       11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004466  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004470  004536                          B3                              ;ROUTINE ADDRESS
004472  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
Z 004474  010020                        MOV     R0,(R0)+                ;DO THE DATO
004476  042777  000004  175116          BIC     #BIT2,@PARITY           ;WRITE NORMAL
004504  011110                          MOV     @R1,@R0                 ;WRITE ADDRESS NORMAL (DATI)
004506  052777  000004  175106          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
004514  012737  004524  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
Z 004522  020030                        CMP     R0,@(R0)+               ;DO A DATI, DATIP
004524  042777  000004  175070          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004532  104001                          HLT     +1                      ;DIDN'T ABORT
004534  000410                          BR      .+22                    ;GO TO NEXT TEST
004536  042777  000005  175056  B3:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004544  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004550  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004552  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*****************************************************************
                                ;TEST 17        TEST (ADDRESS)  SMO,DM3  CMP INSTRUCTION
                                ;*****************************************************************
004556  000004                  TST17:  SCOPE
                                ;                       11/45 **** ROM STATE 221 ****
                                ;
                                ;       11/40 **** ROM STATE 264 ****
004560  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004562  004612                          B4                              ;ROUTINE ADDRESS
004564  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
004566  010010                          MOV     R0,@R0                  ;DO THE DATO
004570  012737  004600  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK.
                                                                        ;IF A PARITY ABORT OCCURS
Z 004576  020030                        CMP     R0,@(R0)+               ;DO A DATI
004600  042777  000004  175014          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004606  104001                          HLT     +1                      ;DIDN'T ABORT
004610  000410                          BR      .+22                    ;GO TO NEXT TEST
004612  042777  000005  175002  B4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004620  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004624  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004626  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*****************************************************************
                                ;TEST 20        TEST DATI (DATA)  SMO,DM5  CMP INSTRUCTION
                                ;*****************************************************************
004632  000004                  TST20:  SCOPE
                                ;                       11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004634  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004636  004706                          B5                              ;ROUTINE ADDRESS
```

```
 1934  004640  011100                        MOV     @R1,R0          ;SET UP FOR A DATO
 1935  004642  042777  000004  174752        BIC     #BIT2,@PARITY   ;WRITE NORMAL
 1936  004650  010060  177776                MOV     R0,-2(R0)       ;DO THE DATO
 1937  004654  052777  000004  174740        BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
 1938  004662  011110                        MOV     @R1,@R0         ;DO THE DATO
 1939  004664  012737  004674  001332        MOV     #.+10,@#$GDDAT  ;STORE THE PC THAT SHOULD
 1940                                                                ;BE PUSHED ON THE STACK
 1941                                                                ;IF A PARITY ABORT OCCURS
Z1942  004672  020050                        CMP     R0,@-(R0)       ;DO A DATI, DATIP
 1943  004674  042777  000004  174720        BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
 1944  004702  104001                        HLT     +1              ;DIDN'T ABORT
 1945  004704  000410                        BR      .+22            ;GO TO NEXT TEST
 1946  004706  042777  000005  174706  B5:   BIC     #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
 1947  004714  004037  011550                JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
 1948  004720  104003                        HLT     +3              ;ABORTED INCORRECTLY
 1949  004722  012706  001100                MOV     #STACK,SP       ;RESET THE STACK
 1950                          ;*************************************************************
 1951                          ;TEST 21        TEST (ADDRESS)  SM0,DM5  CMP INSTRUCTION
 1952                          ;*************************************************************
 1953  004726  000004          TST21:  SCOPE
 1954                          ;                       11/45 **** ROM STATE 231 ****
 1955                          ;
 1956                          ;       11/40 **** ROM STATE 264 ****
 1957  004730  004015                        JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
 1958  004732  004766                        B6                      ;ROUTINE ADDRESS
 1959  004734  011100                        MOV     @R1,R0          ;SET UP FOR A DATO
 1960  004736  010010                        MOV     R0,(R0)         ;DO THE DATO
 1961  004740  062700  000002                ADD     #2,R0
 1962  004744  012737  004754  001332        MOV     #.+10,@#$GDDAT  ;STORE THE PC THAT SHOULD
 1963                                                                ;BE PUSHED ON THE STACK
 1964                                                                ;IF A PARITY ABORT OCCURS
Z1965  004752  020050                        CMP     R0,@-(R0)       ;DO A DATI
 1966  004754  042777  000004  174640        BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
 1967  004762  104001                        HLT     +1              ;DIDN'T ABORT
 1968  004764  000410                        BR      .+22            ;GO TO NEXT TEST
 1969  004766  042777  000005  174626  B6:   BIC     #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
 1970  004774  004037  011550                JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
 1971  005000  104003                        HLT     +3              ;ABORTED INCORRECTLY
 1972  005002  012706  001100                MOV     #STACK,SP       ;RESET THE STACK
 1973                          ;*************************************************************
 1974                          ;TEST 22        TEST (DATA)  SM2,DM0  CMP INSTRUCTION
 1975                          ;*************************************************************
 1976  005006  000004          TST22:  SCOPE
 1977                          ;                       11/45 **** ROM STATE 27 ****
 1978                          ;
 1979                          ;       11/40 **** ROM STATE 250 ****
 1980  005010  004015                        JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
 1981  005012  005042                        C                       ;ROUTINE ADDRESS
 1982  005014  011100                        MOV     @R1,R0          ;SET UP FOR DATO
 1983  005016  010010                        MOV     R0,@R0          ;DO THE DATO
 1984  005020  012737  005030  001332        MOV     #.+10,@#$GDDAT  ;STORE THE PC THAT SHOULD
 1985                                                                ;BE PUSHED ON THE STACK
 1986                                                                ;IF A PARITY ABORT OCCURS
 1987  005026  022000                        CMP     (R0)+,R0        ;DO A DATI
 1988  005030  042777  000004  174564        BIC     #BIT2,@PARITY   ;WRITE NORMAL FOR EMT CALL
 1989  005036  104001                        HLT     +1              ;DIDN'T ABORT
```

```
 1990  005040  000410                          BR      .+22                    ;GO TO NEXT TEST
 1991  005042  042777  000005  174552  C:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
 1992  005050  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 1993  005054  104003                          HLT     +3                      ;ABORTED INCORRECTLY
 1994  005056  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
 1995                                  ;*****************************************************************
 1996                                  ;TEST 23        TEST (DATA)  SM4,DM0  CMP INSTRUCTION
 1997                                  ;*****************************************************************
 1998  005062  000004                  TST23:  SCOPE
```

```
 1999                           ;                11/45 **** ROM STATE 27 ****
```

```
  2000
  2001                                  ;
  2002  005064  004015                  ;       11/40 **** ROM STATE 250 ****
                                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
  2003  005066  005116                  CO                              ;ROUTINE ADDRESS
  2004  005070  011100                  MOV     @R1,R0                  ;SET UP FOR DATO
Z 2005  005072  010020                  MOV     R0,(R0)+                ;DO THE DATO
```

```
 2006  005074  012737  005104  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2007                                                                          ;BE PUSHED ON THE STACK
 2008                                                                          ;IF A PARITY ABORT OCCURS
 2009  005102  024000                          CMP     -(R0),R0                ;DO A DATI
 2010  005104  042777  000004  174510          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
 2011  005112  104001                          HLT     +1                      ;DIDN'T ABORT
 2012  005114  000410                          BR      .+22                    ;GO TO NEXT TEST
 2013  005116  042777  000005  174476  CO:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
 2014  005124  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2015  005130  104003                          HLT     +3                      ;ABORTED INCORRECTLY
 2016  005132  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
 2017                                  ;*****************************************************************
 2018                                  ;TEST 24        TEST (DATA)  SM3,DM0  CMP INSTRUCTION
 2019                                  ;*****************************************************************
 2020  005136  000004                  TST24:  SCOPE
 2021                                  ;                       11/45 **** ROM STATE 146 ****
 2022                                  ;
 2023                                  ;       11/40 **** ROM STATE 250 ****
 2024  005140  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2025  005142  005210                          C1                              ;ROUTINE ADDRESS
 2026  005144  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
Z 2027  005146  010020                          MOV     R0,(R0)+                ;DO THE DATO (DATA OTHER PARITY)
 2028  005150  042777  000004  174444          BIC     #BIT2,@PARITY           ;WRITE NORMAL
 2029  005156  011110                          MOV     @R1,@R0                 ;DO A DATO (ADDRESS NORMAL)
 2030  005160  052777  000004  174434          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
 2031  005166  012737  005176  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2032                                                                          ;BE PUSHED ON THE STACK
 2033                                                                          ;IF A PARITY ABORT OCCURS
 2034  005174  023000                          CMP     @(R0)+,R0               ;DO A DATI
 2035  005176  042777  000004  174416          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
 2036  005204  104001                          HLT     +1                      ;DIDN'T ABORT
 2037  005206  000410                          BR      .+22                    ;GO TO NEXT TEST
 2038  005210  042777  000005  174404  C1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
 2039  005216  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
 2040  005222  104003                          HLT     +3                      ;ABORTED INCORRECTLY
 2041  005224  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
 2042                                  ;*****************************************************************
 2043                                  ;TEST 25        TEST (DATA)  SM5,DM0  CMP INSTRUCTION
 2044                                  ;*****************************************************************
 2045  005230  000004                  TST25:  SCOPE
 2046                                  ;                       11/45 **** ROM STATE 146 ****
 2047                                  ;
 2048                                  ;       11/40 **** ROM STATE 250 ****
 2049  005232  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2050  005234  005304                          C2                              ;ROUTINE ADDRESS
 2051  005236  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
 2052  005240  042777  000004  174354          BIC     #BIT2,@PARITY           ;WRITE NORMAL
 2053  005246  010060  177776                  MOV     R0,-2(R0)               ;DO A DATO (ADDRESS NORMAL)
 2054  005252  052777  000004  174342          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
 2055  005260  011110                          MOV     @R1,@R0                 ;DO A DATO (DATA OTHER PARITY)
 2056  005262  012737  005272  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
 2057                                                                          ;BE PUSHED ON THE STACK
 2058                                                                          ;IF A PARITY ABORT OCCURS
 2059  005270  025000                          CMP     @-(R0),R0               ;DO A DATI
 2060  005272  042777  000004  174322          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
 2061  005300  104001                          HLT     +1                      ;DIDN'T ABORT
```

```
2062  005302  000410                        BR      .+22                    ;GO TO NEXT TEST
2063  005304  042777  000005  174310  C2:   BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2064  005312  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2065  005316  104003                        HLT     +3                      ;ABORTED INCORRECTLY
2066  005320  012706  001100                MOV     #STACK,SP               ;RESET THE STACK
2067                                ;*************************************************************
2068                                ;TEST 26        TEST (DATA)  SM1,DM0  CMP INSTRUCTION
2069                                ;*************************************************************
2070  005324  000004                TST26:  SCOPE
2071                                ;                       11/45 **** ROM STATE 27 ****
2072                                ;
2073                                ;       11/40 **** ROM STATE 250 ****
2074  005326  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2075  005330  005360                        C3                              ;ROUTINE ADDRESS
2076  005332  011100                        MOV     @R1,R0                  ;SET UP FOR DATO
2077  005334  010010                        MOV     R0,@R0                  ;DO THE DATO
2078  005336  012737  005346  001332        MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
2079                                                                        ;BE PUSHED ON THE STACK
2080                                                                        ;IF A PARITY ABORT OCCURS
2081  005344  021000                        CMP     @R0,R0                  ;DO A DATI
2082  005346  042777  000004  174246        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2083  005354  104001                        HLT     +1                      ;DIDN'T ABORT
2084  005356  000410                        BR      .+22                    ;GO TO NEXT TEST
2085  005360  042777  000005  174234  C3:   BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2086  005366  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2087  005372  104003                        HLT     +3                      ;ABORTED INCORRECTLY
2088  005374  012706  001100                MOV     #STACK,SP               ;RESET THE STACK
2089                                ;*************************************************************
2090                                ;TEST 27        TEST (DATA)  SM6,DM0  CMP INSTRUCTION
2091                                ;*************************************************************
2092  005400  000004                TST27:  SCOPE
2093                                ;                       11/45 **** ROM STATE 142 ****
2094                                ;
2095                                ;       11/40 **** ROM STATE 250 ****
2096  005402  004015                        JSR R0,(R5)                     ;SET UP PARITY VECTOR SERVICE
2097  005404  005436                        C4                              ;ROUTINE ADDRESS
2098  005406  011100                        MOV     @R1,R0                  ;SET UP FOR A DATO
Z 2099  005410  010020                        MOV     R0,(R0)+                ;DO THE DATO
2100  005412  012737  005424  001332        MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
2101                                                                        ;BE PUSHED ON THE STACK
2102                                                                        ;IF A PARITY ABORT OCCURS
2103  005420  026000  177776                CMP     -2(R0),R0               ;DO A DATI
2104  005424  042777  000004  174170        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2105  005432  104001                        HLT     +1                      ;DIDN'T ABORT
2106  005434  000410                        BR      .+22                    ;GO TO NEXT TEST
2107  005436  042777  000005  174156  C4:   BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2108  005444  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2109  005450  104003                        HLT     +3                      ;ABORTED INCORRECTLY
2110  005452  012706  001100                MOV     #STACK,SP               ;RESET THE STACK
2111                                ;*************************************************************
2112                                ;TEST 30        TEST (ADDRESS)  SM7,DM0  CMP INSTRUCTION
2113                                ;*************************************************************
2114  005456  000004                TST30:  SCOPE
2115                                ;                       11/45 **** ROM STATE 142 ****
2116                                ;
2117                                ;       11/40 **** ROM STATE 245 ****
```

```
                                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
                                        C5                              ;ROUTINE ADDRESS
                                        MOV     @R1,R0                  ;SET UP FOR DATO
Z                                       MOV     R0,(R0)+                ;DO THE DATO
                                        MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
                                        CMP     @-2(R0),R0              ;DO A DATI
                                        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
                                        HLT     +1                      ;DIDN'T ABORT
                                        BR      .+22                    ;GO TO NEXT TEST
                                C5:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
                                        JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
                                        HLT     +3                      ;ABORTED INCORRECTLY
                                        MOV     #STACK,SP               ;RESET THE STACK
                                ;***************************************************************
                                ;TEST 31        TEST (ADDRESS)  SM3,DM0  CMP INSTRUCTION
                                ;***************************************************************
                                TST31:  SCOPE
                                ;                       11/45 **** ROM STATE 27 ****
                                ;
                                ;       11/40 **** ROM STATE 245 ****
                                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
                                        C6                              ;ROUTINE ADDRESS
                                        MOV     @R1,R0                  ;SET UP FOR A DATO
                                        MOV     R0,@R0                  ;DO THE DATO
                                        MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
                                        CMP     @(R0)+,R0               ;DO A DATI
                                        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
                                        HLT     +1                      ;DIDN'T ABORT
                                        BR      .+22                    ;GO TO NEXT TEST
                                C6:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
                                        JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
                                        HLT     +3                      ;ABORTED INCORRECTLY
                                        MOV     #STACK,SP               ;RESET THE STACK
                                ;***************************************************************
                                ;TEST 32        TEST (ADDRESS)  SM5,DM0  CMP INSTRUCTION
                                ;***************************************************************
                                TST32:  SCOPE
                                ;                       11/45 **** ROM STATE 27 ****
                                ;
                                ;       11/40 **** ROM STATE 245 ****
                                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
                                        C7                              ;ROUTINE ADDRESS
                                        MOV     @R1,R0                  ;SET UP FOR A DATO
Z                                       MOV     R0,(R0)+                ;DO THE DATO
                                        MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
                                        CMP     @-(R0),R0               ;DO A DATI
                                        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
                                        HLT     +1                      ;DIDN'T ABORT
                                        BR      .+22                    ;GO TO NEXT TEST
                                C7:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
```

```
2174  005652  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2175  005656  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2176  005660  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
2177                                  ;*****************************************************************
2178                                  ;TEST 33        TEST (DATA)  SM7,DM0  CMP INSTRUCTION
2179                                  ;*****************************************************************
2180  005664  000004                  TST33:  SCOPE
2181                                  ;                       11/45 **** ROM STATE 146 ****
2182                                  ;
2183                                  ;       11/40 **** ROM STATE 250 ****
2184  005666  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2185  005670  005742                          C8                              ;ROUTINE ADDRESS
2186  005672  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
2187  005674  042777  000004  173720          BIC     #BIT2,@PARITY           ;WRITE NORMAL
2188  005702  010060  177776                  MOV     R0,-2(R0)               ;DO A DATO (ADDRESS NORMAL)
2189  005706  052777  000004  173706          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
2190  005714  011110                          MOV     @R1,@R0                 ;DO A DATO (DATA OTHER PARITY)
2191  005716  012737  005730  001332          MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
2192                                                                          ;BE PUSHED ON THE STACK
2193                                                                          ;IF A PARITY ABORT OCCURS
2194  005724  027000  177776                  CMP     @-2(R0),R0              ;DO A DATI
2195  005730  042777  000004  173664          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2196  005736  104001                          HLT     +1                      ;DIDN'T ABORT
2197  005740  000410                          BR      .+22                    ;GO TO NEXT TEST
2198  005742  042777  000005  173652  C8:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2199  005750  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2200  005754  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2201  005756  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
2202                                  ;*****************************************************************
2203                                  ;TEST 34        TEST  DM3  JMP INSTRUCTION
2204                                  ;*****************************************************************
2205  005762  000004                  TST34:  SCOPE
2206                                  ;                       11/45 **** ROM STATE 221 ****
2207                                  ;
2208                                  ;       11/40 **** ROM STATE 303 ****
2209  005764  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2210  005766  006020                          D                               ;ROUTINE ADDRESS
2211  005770  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
2212  005772  012710  006006                  MOV     #DD,@R0                 ;DO THE DATO
2213  005776  012737  006006  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
2214                                                                          ;BE PUSHED ON THE STACK
2215                                                                          ;IF A PARITY ABORT OCCURS
2216  006004  000130                          JMP     @(R0)+                  ;DO A DATI
2217  006006  042777  000004  173606  DD:     BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2218  006014  104001                          HLT     +1                      ;DIDN'T ABORT
2219  006016  000410                          BR      .+22                    ;GO TO NEXT TEST
2220  006020  042777  000005  173574  D:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2221  006026  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2222  006032  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2223  006034  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
2224                                  ;*****************************************************************
2225                                  ;TEST 35        TEST  DM5  JMP INSTRUCTION
2226                                  ;*****************************************************************
2227  006040  000004                  TST35:  SCOPE
2228                                  ;                       11/45 **** ROM STATE 231 ****
2229                                  ;
```

```
                                ;       11/40 **** ROM STATE 303 ****
006042  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
006044  006076                          D0                              ;ROUTINE ADDRESS
006046  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
006050  012720  006064                  MOV     #DD0,(R0)+              ;DO THE DATO
006054  012737  006064  001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
006062  000150                          JMP     @-(R0)                  ;DO A DATI
006064  042777  000004  173530  DD0:    BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
006072  104001                          HLT     +1                      ;DIDN'T ABORT
006074  000410                          BR      .+22                    ;GO TO NEXT TEST
006076  042777  000005  173516  D0:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
006104  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
006110  104003                          HLT     +3                      ;ABORTED INCORRECTLY
006112  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*******************************************************************
                                ;TEST 36        TEST  DM7  JMP INSTRUCTION
                                ;*******************************************************************
006116  000004                  TST36:  SCOPE
                                ;                       11/45 **** ROM STATE 231 ****
                                ;
                                ;       11/40 **** ROM STATE 303 ****
006120  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
006122  006156                          D1                              ;ROUTINE ADDRESS
006124  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
006126  012720  006144                  MOV     #DD1,(R0)+              ;DO THE DATO
006132  012737  006144  001332          MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
006140  000170  177776                  JMP     @-2(R0)                 ;DO A DATI
006144  042777  000004  173450  DD1:    BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
006152  104001                          HLT     +1                      ;DIDN'T ABORT
006154  000410                          BR      .+22                    ;GO TO NEXT TEST
006156  042777  000005  173436  D1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
006164  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
006170  104003                          HLT     +3                      ;ABORTED INCORRECTLY
006172  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK


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

                                ;*****************************************************************
                                ;TEST 37        TEST   MARK "POP"   MARK INSTRUCTION
                                ;*****************************************************************
006176  000004                  TST37:  SCOPE
                                ;
                                ;               11/45 **** ROM STATE 260 ****
                                ;
                                ;       11/40 **** ROM STATE 1 ****
006200  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
006202  006326                          EO                              ;ROUTINE ADDRESS
006204  042777  000004  173410          BIC     #BIT2,@PARITY           ;WRITE NORMAL
                                ;*****************************************************************
                                ;*****************************************************************
                                ;*****************************************************************
                                ;*****************************************************************
                                ;
                                ;NOTE THAT THE NEXT INSTRUCTION WILL MOVE THE STACK TO PARITY AREA
                                ;IN THE EVENT THAT NO PARITY EXISTS WHERE THE STACK IS NORMALLY
                                ;SITTING AT 1100
                                ;
                                ;*****************************************************************
                                ;*****************************************************************
                                ;*****************************************************************
                                ;*****************************************************************
006212  013706  001476                  MOV     @#NEWSTK,SP             ;SET THE STACK IN PARITY
                                                                        ;MEMORY AREA
006216  017702  173224                  MOV     @NTERAD,R2              ;GET THE INTERLEAVE FACTOR
                                                                        ;FOR THIS CONTROLER
006222  005302                          DEC     R2                      ;CALCULATE NO. OF PARAMETERS
                                                                        ;TO BE PUSHED ON THE STACK
006224  010546                          MOV     R5,-(SP)                ;PUSH R5 CONTENTS ON STACK
006226  005702                  1$:     TST     R2                      ;ANY PARAMETERS TO BE
                                                                        ;PUSHED ON THE STACK?
```

```
2342  006230  001404                          BEQ     2$              ;BRANCH IF NO
2343  006232  005302                          DEC     R2              ;SUBTRACT 1 FROM PARAMETER COUNT
2344  006234  012746  000001                  MOV     #1,-(SP)        ;PUSH PARAMETER ONTO STACK
2345  006240  000772                          BR      1$              ;GO BACK TO SEE IF ANY MORE!
2346  006242  052777  000004  173352  2$:     BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
2347  006250  017702  173172                  MOV     @NTERAD,R2      ;GET THE INTERLEAVE FACTOR
2348                                                                  ;FOR THIS CONTROLLER
2349  006254  005302                          DEC     R2              ;CALCULATE NO. OF PARAMETERS
2350                                                                  ;THAT WERE TO BE PUSHED ON THE
2351                                                                  ;STACK
2352  006256  062702  006400                  ADD     #6400,R2        ;CALCULATE THE CORRESPONDING
2353                                                                  ;MARK INSTRUCTION
2354  006262  010246                          MOV     R2,-(SP)        ;PUSH MARK INSTRUCTION ON STACK
2355  006264  042777  000004  173330          BIC     #BIT2,@PARITY   ;WRITE NORMAL
2356  006272  010605                          MOV     SP,R5           ;PLACE MARK INSTRUCTION ADDRESS
2357                                                                  ;INTO R5 FOR RTS
2358  006274  010602                          MOV     SP,R2           ;GET THE PC OF THE MARK
2359                                                                  ;INSTRUCTION
2360  006276  005737  001642                  TST     @#CPU40         ;ARE WE ON AN 11/40?
2361  006302  001002                          BNE     3$              ;BRANCH IF YES
2362                                                                  ;AND DON'T STEP UP THE PC
2363                                                                  ;IT WON'T BE UPDATED ON THE
2364                                                                  ;PARITY ABORT
2365  006304  062702  000002                  ADD     #2,R2           ;STEP UP THE PC
2366                                                                  ;IT WILL BE UPDATED ON THE
2367                                                                  ;PARITY ABORT
2368  006310  010237  001332          3$:     MOV     R2,@#$GDDAT     ;STORE THIS VALUE OF PC
2369                                                                  ;THAT SHOULD BE PUSHED ON THE
2370                                                                  ;STACK IF A PARITY ABORT OCCURS
2371  006314  004767  000004                  JSR     PC,MRK0         ;SUBROUTINE CALL
2372  006320  104001                  E:      HLT     +1              ;DIDN'T ABORT
2373  006322  000407                          BR      .+20            ;GO TO RESET THE STACK
2374  006324  000205                  MRK0:   RTS     R5              ;RETURN FROM SUBROUTINE
2375  006326  042777  000001  173266  E0:     BIC     #BIT0,@PARITY   ;DISABLE PARITY
2376  006334  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
2377  006340  104003                          HLT     +3              ;ABORTED INCORRECTLY
2378  006342  013706  001476                  MOV     @#NEWSTK,SP     ;RESET THE STACK
2379
2380
2381
```

```
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
;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
                        ;
                        ;*****************************************************************
                        ;*****************************************************************
                        ;*****************************************************************
                        ;*****************************************************************
 2416
 2417
 2418
 2419  006346  012705  011450                MOV     #VECSET,R5              ;RESTORE THE PARITY VECTOR
 2420                                                                          ;SERVICE ADDRESS SETUP ROUTINE
 2421                                                                          ;ADDRESS
 2422                                 ;*****************************************************************
 2423                                 ;TEST 40        TEST   OLD REGISTER CONTENTS   MARK INSTRUCTION
 2424                                 ;*****************************************************************
 2425  006352  000004                TST40:  SCOPE
 2426                                 ;
 2427                                 ;       11/45 **** ROM STATE 235 ****
 2428                                 ;
 2429  006354  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
 2430  006356  006510                        E2                              ;ROUTINE ADDRESS
 2431  006360  017702  173062                MOV     @NTERAD,R2              ;GET THE INTERLEAVE FACTOR
 2432                                                                          ;FOR THIS CONTROLLER
 2433  006364  005302                        DEC     R2                      ;CALCULATE NO. OF PARAMETERS
 2434                                                                          ;TO BE PUSHED ON THE STACK
 2435  006366  042777  000004  173226        BIC     #BIT2,@PARITY           ;WRITE NORMAL
 2436  006374  005737  001642                TST     @#CPU40                 ;ARE WE ON AN 11/40?
 2437  006400  001407                        BEQ     3$                      ;BRANCH IF NO
 2438                                                                          ;DATA WILL BE TAKEN FIRST AND
 2439                                                                          ;THEN THE PC UPDATED
 2440  006402  013700  001476                MOV     @#NEWSTK,R0             ;GET THE INITIAL STACK POINT
 2441  006406  162700  000002                SUB     #2,R0                   ;DROP DOWN 1 WORD ADDRESS
 2442                                                                          ;BECAUSE THE PC WILL BE UPDATED
 2443                                                                          ;FIRST THEN THE DATA TAKEN
 2444  006412  010037  001332                MOV     R0,@#$GDDAT             ;STORE THIS VALUE AS THE PC
 2445                                                                          ;THAT SHOULD BE PUSHED ON THE STACK
 2446                                                                          ;IF A PARITY ABORT OCCURS
 2447  006416  000403                        BR      4$                      ;CONTINUE WITH TEST
 2448  006420  012737  006502  001332  3$:   MOV     #E1,@#$GDDAT            ;STORE THE PC THAT SHOULD
 2449                                                                          ;BE PUSHED ON THE STACK
 2450                                                                          ;IF A PARITY ABORT OCCURS
 2451  006426  052777  000004  173166  4$:   BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
 2452  006434  010546                        MOV     R5,-(SP)                ;STORE OLD R5 CONTENTS ON STACK
 2453  006436  042777  000004  173156        BIC     #BIT2,@PARITY           ;WRITE NORMAL
```

```
2454  006444  005702                    1$:     TST     R2              ;ANY PARAMETERS TO BE PUSHED
2455                                                                    ;ON THE STACK?
2456  006446  001404                            BEQ     2$              ;BRANCH IF NO
2457  006450  005302                            DEC     R2              ;SUBTRACT 1 FROM PARAMETER COUNT
2458  006452  012746  000001                    MOV     #1,-(SP)        ;PUSH PARAMETER ON STACK
2459  006456  000772                            BR      1$              ;GO BACK TO SEE IF ANY MORE!
2460  006460  017702  172762            2$:     MOV     @NTERAD,R2      ;GET THE INTERLEAVE FACTOR
2461                                                                    ;FOR THIS CONTROLLER
2462  006464  005302                            DEC     R2              ;CALCULATE NO. OF PARAMETERS
2463                                                                    ;THAT WERE TO BE PUSHED ON THE
2464                                                                    ;STACK
2465  006466  062702  006400                    ADD     #6400,R2        ;CALCULATE THE CORRESPONDING
2466                                                                    ;MARK INSTRUCTION
2467  006472  010246                            MOV     R2,-(SP)        ;PUSH MARK INSTRUCTION ON STACK
2468  006474  010605                            MOV     SP,R5           ;PLACE MARK INSTRUCTION ADDRESS
2469                                                                    ;INTO R5 FOR RTS
2470  006476  004767  000004                    JSR     PC,MRK1         ;SUBROUTINE CALL
2471  006502  104001                    E1:     HLT     +1              ;DIDN'T ABORT
2472  006504  000407                            BR      .+20            ;GO TO RESET THE STACK
2473  006506  000205                    MRK1:   RTS     R5              ;RETURN FROM SUBROUTINE
2474  006510  042777  000001  173104    E2:     BIC     #BIT0,@PARITY   ;DISABLE PARITY
2475  006516  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
2476  006522  104003                            HLT     +3              ;ABORTED INCORRECTLY
2477  006524  013706  001476                    MOV     @#NEWSTK,SP     ;RESET THE STACK
2478  006530  012705  011450                    MOV     #VECSET,R5      ;RESTORE THE PARITY VECTOR
2479                                                                    ;SERVICE ADDRESS SETUP ROUTINE
2480                                                                    ;ADDRESS
2481                              ;*****************************************************************
2482                              ;TEST 41        TEST   SOB BRANCH   SOB INSTRUCTION
2483                              ;*****************************************************************
2484  006534  000004              TST41:  SCOPE
2485                                      ;               11/45 **** ROM STATE 260 ****
2486                                      ;
2487                                      ;       11/40 **** ROM STATE 1 ****
2488  006536  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
2489  006540  006644                            FD                      ;ROUTINE ADDRESS
2490  006542  042777  000004  173052            BIC     #BIT2,@PARITY   ;WRITE NORMAL
2491  006550  012700  000004                    MOV     #4,R0           ;MOVE A NUMBER >1 TO R0 SO THAT
2492                                                                    ;WHEN DECREMENTED BY "SOB" THE
2493                                                                    ;RESULT WON'T BE 0
2494  006554  011102                            MOV     @R1,R2          ;SET UP FOR A DATO
2495  006556  013703  001476                    MOV     @#NEWSTK,R3     ;GET THE INITIAL TEST POINT
2496  006562  005737  001642                    TST     @#CPU40         ;ARE WE ON AN 11/40?
2497  006566  001403                            BEQ     1$              ;BRANCH IF NO
2498  006570  062703  000002                    ADD     #2,R3           ;STEP UP 1 WORD
2499                                                                    ;SINCE THE PC WILL NOT BE
2500                                                                    ;UPDATED ON THE PARITY ABORT
2501  006574  000402                            BR      2$              ;GO TO STORE THIS VALUE
2502  006576  062703  000004            1$:     ADD     #4,R3           ;STEP UP 2 WORDS
2503                                                                    ;SINCE THE PC WILL BE UPDATED
2504                                                                    ;ON THE PARITY ABORT
2505  006602  010337  001332            2$:     MOV     R3,@#$GDDAT     ;STORE THIS VALUE OF PC
2506                                                                    ;THAT SHOULD BE PUSHED ON THE
2507                                                                    ;STACK IF A PARITY ABORT OCCURS
2508  006606  052777  000004  173006            BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
2509  006614  012722  000240                    MOV     #240,(R2)+      ;INSTRUCTION TO BE DONE ON
```

```
2510                                                    ;COMPLETION OF SOB INSTRUCTION
2511                                                    ;THE INSTRUCTION WILL BE IN
2512                                                    ;PARITY MEMORY AS DESIGNATED BY
2513                                                    ;THE CONTENTS OF R1
2514  006620  042777  000004  172774          BIC     #BIT2,@PARITY     ;WRITE NORMAL
2515  006626  012712  000203                  MOV     #203,@R2          ;MOVE AN RTS R3 INTO THE LOCATION
2516                                                    ;FOLLOWING THE INSTRUCTION TO BE
2517                                                    ;EXECUTED IN PARITY MEMORY IN
2518                                                    ;CASE IT DOESN'T ABORT
2519  006632  000403                  BR      .+10      ;GO TO SOB INSTRUCTION
2520  006634  004342          F:      JSR     R3,-(R2)  ;THE INSTRUCTION TO BE DONE ON
2521                                                    ;COMPLETION OF SOB EXECUTION IS
2522                                                    ;PLACED HERE
2523  006636  104001                  HLT     +1        ;DIDN'T ABORT
2524  006640  000407                  BR      .+20      ;GO TO RESET THE STACK
2525  006642  077004                  SOB     R0,F      ;EXECUTE THIS INSTRUCTION
2526  006644  042777  000001  172750  F0:     BIC     #BIT0,@PARITY     ;DISABLE PARITY
2527  006652  004037  011550          JSR     R0,@#CHECKLOC     ;CHECK FOR GOOD ABORT
2528  006656  104003                  HLT     +3        ;ABORTED INCORRECTLY
2529  006660  013706  001476          MOV     @#NEWSTK,SP       ;RESET THE STACK
2530                          ;*****************************************************************
2531                          ;TEST 42        TEST  SMO,DMO  MOV INSTRUCTION
2532                          ;*****************************************************************
2533  006664  000004         TST42:  SCOPE
2534                          ;                       11/45 **** ROM STATE 20 ****
2535                          ;
2536                          ;       11/40 **** ROM STATE 1 ****
2537  006666  004015                  JSR     R0,(R5)           ;SET UP PARITY VECTOR SERVICE
2538  006670  006774                  F1                        ;ROUTINE ADDRESS
2539  006672  011100                  MOV     @R1,R0            ;SET UP FOR A DATO
2540  006674  017702  172546          MOV     @INTERAD,R2       ;GET THE INTERLEAVE FACTOR
2541                                                    ;FOR THIS CONTROLLER
2542  006700  042777  000004  172714          BIC     #BIT2,@PARITY     ;WRITE NORMAL
2543  006706  005702          1$:     TST     R2        ;ANY 'MOV R0,R0'S TO BE SHOVED
2544                                                    ;INTO PARITY MEMORY AREA?
2545  006710  001404                  BEQ     2$        ;BRANCH IF NO
2546  006712  005302                  DEC     R2        ;SUBTRACT 1 FROM INSTRUCTION
2547                                                    ;COUNT
2548  006714  012720  010000          MOV     #010000,(R0)+     ;MOVE THE INSTRUCTION 'MOV R0,R0'
2549                                                    ;TO PARITY AREA
2550  006720  000772                  BR      1$        ;GO BACK TO SEE IF ANY MORE!
2551  006722  052777  000004  172672  2$:     BIS     #BIT2,@PARITY     ;WRITE OTHER PARITY
2552  006730  012720  010000          MOV     #010000,(R0)+     ;MOVE THE INSTRUCTION 'MOV R0,R0'
2553                                                    ;INTO THE NEXT WORD LOCATION AFTER
2554                                                    ;THE PREVIOUS 'MOV R0,R0' INSTRUCTION
2555  006734  042777  000004  172660          BIC     #BIT2,@PARITY     ;WRITE NORMAL
2556  006742  012710  000203                  MOV     #203,(R0)         ;MOVE 'RTS R3' INTO NEXT WORD
2557                                                    ;LOCATION AFTER THE PREVIOUS
2558                                                    ;'MOV R0,R0' INSTRUCTION IN CASE
2559                                                    ;A PARITY ABORT DOESN'T OCCUR
2560  006746  005737  001642          TST     @#CPU40   ;ARE WE ON AN 11/40?
2561  006752  001402                  BEQ     3$        ;BRANCH IF NO
2562                                                    ;SINCE PC WILL BE UPDATED ON THE
2563                                                    ;PARITY ABORT
2564  006754  162700  000002          SUB     #2,R0     ;DROP BACK 1 WORD ADDRESS
2565                                                    ;SINCE THE PC WILL NOT BE
```

```
2566
2567  006760  010037  001332          3$:     MOV     R0,@#$GDDAT     ;UPDATED ON THE PARITY ABORT
2568                                              ;STORE THE PC THAT SHOULD
2569                                              ;BE PUSHED ON THE STACK
2570  006764  004371  000000                    JSR     R3,@(R1)        ;IF A PARITY ABORT OCCURS
2571  006770  104001                              HLT     +1              ;GO TO PARITY MEMORY AREA
2572  006772  000406                              BR      .+16            ;DIDN'T ABORT
2573  006774  042777  000001  172620  F1:       BIC     #BIT0,@PARITY   ;GO TO RESET THE STACK
2574  007002  004037  011550                    JSR     R0,@#CHECKLOC   ;DISABLE PARITY
2575  007006  104003                              HLT     +3              ;CHECK FOR GOOD ABORT
2576  007010  013706  001476                    MOV     @#NEWSTK,SP     ;ABORTED INCORRECTLY
2577                                              ;RESET THE STACK
2578
2579

        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
        ;
        ;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)
        ;
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************

2598
2599
2600                                    ;*************************************************************************
2601                                    ;TEST 43        TEST   RTS INSTRUCTION
2602                                    ;*************************************************************************
2603  007014  000004                    TST43:  SCOPE
2604                                    ;
2605                                    ;               11/45 **** ROM STATE 224 ****
2606                                    ;
2607  007016  004015                    ;       11/40 **** ROM STATE 325 ****
2608  007020  007066                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
2609  007022  042777  000004  172572            GO                      ;ROUTINE ADDRESS
2610  007030  012700  007056                    BIC     #BIT2,@PARITY   ;WRITE NORMAL
2611  007034  012737  007052  001332            MOV     #G,R0           ;SET UP A SUBROUTINE ADDRESS
2612                                              MOV     #.+16,@#$GDDAT  ;STORE THE PC THAT SHOULD
2613                                                                      ;BE PUSHED ON THE STACK
2614  007042  052777  000004  172552                                    ;IF A PARITY ABORT OCCURS
Z 2615  007050  004320                            BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
2616  007052  104001                              JSR     R3,(R0)+        ;SUBROUTINE CALL
2617  007054  000412                              HLT     +1              ;DIDN'T ABORT
2618  007056  042777  000004  172536  G:         BR      .+26            ;GO TO RESET THE STACK
2619  007064  000203                              BIC     #BIT2,@PARITY   ;WRITE NORMAL
2620  007066  042777  000001  172526  GO:       RTS     R3              ;RETURN FROM SUBROUTINE
2621  007074  004037  011550                    BIC     #BIT0,@PARITY   ;DISABLE PARITY
                                                JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
```

```
2622  007100  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2623  007102  013706  001476                  MOV     @#NEWSTK,SP             ;RESET THE STACK
2624
2625
2626

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

        ;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'
        ;
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
2656
2657
2658                                    ;*************************************************************************
2659                                    ;TEST 44        TEST   1ST POP   RTI INSTRUCTION
2660                                    ;*************************************************************************
2661  007106  000004                    TST44:  SCOPE
2662                                    ;                       11/45 **** ROM STATE 212 ****
2663                                    ;
2664                                    ;       11/40 **** ROM STATE 320 ****
2665  007110  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2666  007112  007150                            H0                              ;ROUTINE ADDRESS
2667  007114  012746  000340                    MOV     #340,-(SP)              ;SET PS FOR 'RTI'
2668  007120  012746  007134                    MOV     #H1,-(SP)               ;SET RETURN FROM 'RTI'
2669  007124  012737  007134  001332            MOV     #H1,@#$GDDAT            ;STORE THE PC THAT SHOULD
2670                                                                            ;BE PUSHED ON THE STACK
2671                                                                            ;IF A PARITY ABORT OCCURS
2672  007132  000402                            BR      H                       ;GO DO 'RTI'
2673  007134  104001                    H1:     HLT     +1                      ;DIDN'T ABORT
2674  007136  000412                            BR      .+26                    ;GO TO RESET THE STACK
2675  007140  042777  000004  172454    H:      BIC     #BIT2,@PARITY           ;WRITE NORMAL
2676  007146  000002                            RTI                             ;RETURN FROM INTERRUPT
2677  007150  042777  000001  172444    H0:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
```

```
2678  007156  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2679  007162  104003                          HLT     +3                      ;ABORTED INCORRECTLY
2680  007164  013706  001476                  MOV     @#NEWSTK,SP             ;RESET THE STACK
2681
2682
2683
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************

        ;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)
        ;
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
        ;*************************************************************************
2714
2715
2716                                    ;*************************************************************
2717                                    ;TEST 45        TEST   2ND POP   RTI INSTRUCTION
2718                                    ;*************************************************************
2719  007170  000004                    TST45:  SCOPE
2720                                    ;                       11/45 **** ROM STATE 214 ****
2721                                    ;
2722                                    ;       11/40 **** ROM STATE 322 ****
2723  007172  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2724  007174  007242                            H2                              ;ROUTINE ADDRESS
2725  007176  012746  000340                    MOV     #340,-(SP)              ;SET PS FOR 'RTI'
2726  007202  012746  007216                    MOV     #H3,-(SP)               ;SET RETURN FROM 'RTI'
2727  007206  012737  007216  001332            MOV     #H3,@#$GDDAT            ;STORE THE PC THAT SHOULD
2728                                                                            ;BE PUSHED ON THE STACK
2729                                                                            ;IF A PARITY ABORT OCCURS
2730  007214  000402                            BR      H4                      ;GO DO 'RTI'
2731  007216  104001                    H3:     HLT     +1                      ;DIDN'T ABORT
2732  007220  000416                            BR      .+36                    ;GO TO RESET THE STACK
2733  007222  042777  000005  172372    H4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE PARITY
```

```
                                        MOV     (SP)+,-(SP)             ;WRITE 1ST ''POP'' NORMAL SO NO
                                                                        ;ERROR WILL OCCUR
                                        BIS     #BIT0,@PARITY           ;ENABLE PARITY
                                        RTI                             ;RETURN FROM INTERRUPT
                                H2:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
                                        JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
                                        HLT     +3                      ;ABORTED INCORRECTLY
                                        MOV     #STACK,SP               ;INITIALIZE THE STACK BACK
                                                                        ;TO 1100 FOR WE DON'T NEED IT
                                                                        ;TO BE IN PARITY MEMORY AREA
                                                                        ;ANYMORE
                                ;***************************************************************
                                ;TEST 46        TEST (DATA)   SM6,DM0   ADD INSTRUCTION
                                ;***************************************************************
                                TST46:  SCOPE
                                ;                       11/45 **** ROM STATE 142 ****
                                ;
                                ;       11/40 **** ROM STATE 250 ****
                                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
                                        L                               ;ROUTINE ADDRESS
                                        MOV     @R1,R0                  ;SET UP FOR A DATO
Z                               007272  010020  MOV     R0,(R0)+        ;DO THE DATO
                        007306  001332  MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
                        177776          ADD     -2(R0),R0               ;DO A DATI
                        172306          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
                                        HLT     +1                      ;DIDN'T ABORT
                                        BR      .+22                    ;GO TO NEXT TEST
                        172274  L:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
                        011550          JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
                                        HLT     +3                      ;ABORTED INCORRECTLY
                        001100          MOV     #STACK,SP               ;RESET THE STACK
                                ;***************************************************************
                                ;TEST 47        TEST  DM1  TSTB INSTRUCTION
                                ;***************************************************************
                                TST47:  SCOPE
                                ;                       11/45 **** ROM STATE 175 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
                                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
                                        M                               ;ROUTINE ADDRESS
                                        MOV     @R1,R0                  ;SET UP FOR A DATO
                                        MOV     R0,(R0)                 ;DO THE DATO
                        001332          MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
                                        TSTB    (R0)                    ;DO A DATI, DATIP
                        172232          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
                                        HLT     +1                      ;DIDN'T ABORT
                                        BR      .+22                    ;GO TO NEXT TEST
                        172220  M:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
                        011550          JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
                                        HLT     +3                      ;ABORTED INCORRECTLY
                        001100          MOV     #STACK,SP               ;RESET THE STACK
                                ;***************************************************************
```

```
2790                                    ;TEST 50        TEST DM3  CLRB INSTRUCTION
2791                                    ;*****************************************************************
2792  007414  000004                    TST50:  SCOPE
2793                                    ;                       11/45 **** ROM STATE 221 ****
2794                                    ;
2795                                    ;       11/40 **** ROM STATE 264 ****
2796  007416  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2797  007420  007450                            NN                              ;ROUTINE ADDRESS
2798  007422  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
2799  007424  010010                            MOV     R0,(R0)                 ;DO THE DATO
2800  007426  012737  007436  001332            MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
2801                                                                            ;BE PUSHED ON THE STACK
2802                                                                            ;IF A PARITY ABORT OCCURS
2803  007434  105030                            CLRB    @(R0)+                  ;DO A DATI
2804  007436  042777  000004  172156            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2805  007444  104001                            HLT     +1                      ;DIDN'T ABORT
2806  007446  000410                            BR      .+22                    ;GO TO NEXT TEST
2807  007450  042777  000005  172144  NN:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2808  007456  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2809  007462  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2810  007464  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2811                                    ;*****************************************************************
2812                                    ;TEST 51        TEST  SM1  SUBTRACT INSTRUCTION
2813                                    ;*****************************************************************
2814  007470  000004                    TST51:  SCOPE
2815                                    ;                       11/45 **** ROM STATE 27 ****
2816                                    ;
2817                                    ;       11/40 **** ROM STATE 250 ****
2818  007472  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2819  007474  007526                            P                               ;ROUTINE ADDRESS
2820  007476  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
2821  007500  010010                            MOV     R0,(R0)                 ;DO THE DATO
2822  007502  012737  007512  001332            MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
2823                                                                            ;BE PUSHED ON THE STACK
2824                                                                            ;IF A PARITY ABORT OCCURS
2825  007510  161027  177777                    SUB     (R0),#177777            ;DO A DATI
2826  007514  042777  000004  172100            BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
2827  007522  104001                            HLT     +1                      ;DIDN'T ABORT
2828  007524  000410                            BR      .+22                    ;GO TO NEXT TEST
2829  007526  042777  000005  172066  P:       BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
2830  007534  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
2831  007540  104003                            HLT     +3                      ;ABORTED INCORRECTLY
2832  007542  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
2833                                    ;*****************************************************************
2834                                    ;TEST 52        TEST (DATA)  SM5  BISB INSTRUCTION
2835                                    ;*****************************************************************
2836  007546  000004                    TST52:  SCOPE
2837                                    ;                       11/45 **** ROM STATE 177 ****
2838                                    ;
2839                                    ;       11/40 **** ROM STATE 245 ****
2840  007550  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
2841  007552  007604                            R                               ;ROUTINE ADDRESS
2842  007554  011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
2843  007556  010010                            MOV     R0,(R0)                 ;DO THE DATO
2844  007560  012737  007570  001332            MOV     #.+10,@#$GDDAT          ;STORE THE PC THAT SHOULD
2845                                                                            ;BE PUSHED ON THE STACK
```

```
2846
2847  007566  155060  000002                  BISB    @-(R0),2(R0)        ;IF A PARITY ABORT OCCURS
2848  007572  042777  000004  172022          BIC     #BIT2,@PARITY       ;DO A DATI
2849  007600  104001                          HLT     +1                  ;WRITE NORMAL FOR EMT CALL
2850  007602  000410                          BR      .+22                ;DIDN'T ABORT
2851  007604  042777  000005  172010  R:      BIC     #BIT2!BIT0,@PARITY  ;GO TO NEXT TEST
2852  007612  004037  011550                  JSR     R0,@#CHECKLOC       ;WRITE NORMAL AND DISABLE
2853  007616  104003                          HLT     +3                  ;CHECK FOR GOOD ABORT
2854  007620  012706  001100                  MOV     #STACK,SP           ;ABORTED INCORRECTLY
2855                                                                      ;RESET THE STACK
                                ;*********************************************************************
                                ;TEST 53        NEXT 4 TESTS ARE NON-INTERLEAVE 4K DEPENDENT
                                ;*********************************************************************
2858  007624  000004            TST53:  SCOPE
2859  007626  022701  017360                  CMP     #17360,R1           ;HAVE WE PARITY IN THE LOWER 4K?
2860                                                                      ;AND IS THE SELECTED REGISTER
2861                                                                      ;GOVERNING THE 4K AREA?
2862                                                                      ;THIS COMPARE IS IF THE KT11
2863                                                                      ;ISN'T ENABLED DURING PROGRAM
2864                                                                      ;EXECUTION
2865  007632  101005                          BHI     2$                  ;YES - PROCEED TO NEXT TESTS
2866                                                                      ;WHICH DEPEND ON PARITY IN LOWER 4K
2867  007634  022777  000140  171554          CMP     #140,@$SETAD        ;THE ABOVE COMPARE DIDN'T
2868                                                                      ;CHECK - - DO THIS COMPARE TO
2869                                                                      ;SEE IF IT WAS BECAUSE THE
2870                                                                      ;TABLE WAS CREATED WITH MEMORY
2871                                                                      ;MANAGEMENT TURNED ON
2872  007642  001401                          BEQ     2$                  ;PROCEED TO NEXT TESTS IF THIS
2873                                                                      ;COMPARE CHECKS
2874  007644  000404                          BR      4$                  ;THE REGISTER UNDER TEST IS NOT
2875                                                                      ;CONTROLLING THE LOWER 4K!!!
2876                                                                      ;GO TO RE-EVALUATE $TSTNM AND
2877                                                                      ;JUMP OVER THE (4) 4K DEPENDENT
2878                                                                      ;TESTS
2879  007646  022777  000001  171572  2$:     CMP     #1,@NTERAD          ;IS THIS CONTROLLER
2880                                                                      ;INTERLEAVED??
2881  007654  001405                          BEQ     3$                  ;BRANCH IF NO
2882  007656  062767  000004  171416  4$:     ADD     #4,$TSTNM           ;SET $TSTNM TO PROPER VALUE
2883                                                                      ;SINCE THE NEXT 4 TESTS WILL
2884                                                                      ;BE SKIPPED
2885  007664  000137  010426                  JMP     @#RED+26            ;JUMP TO 1ST INDEX WORD TEST
2886  007670  005237  001624          3$:     INC     @#PSPCORZONES       ;SET FLAG INDICATING TO 'CHECKLOC'
2887                                                                      ;ROUTINE THAT PS AND PC FETCH
2888                                                                      ;AND RED & YELLOW ZONE AREAS
2889                                                                      ;ARE GOING TO BE TESTED
2890
2891
2892

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

```
                        ;*****************************************************************
                        ;TEST 54        TEST NEW 'PS' FETCH
                        ;*****************************************************************
007674 000004           TST54:  SCOPE
                        ;                       11/45 **** ROM STATE 357 ****
                        ;
                        ;       11/40 **** ROM STATE 115 ****
007676 004015                   JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
007700 007754                   TO                              ;ROUTINE ADDRESS
007702 042777 000004 171712     BIC     #BIT2,@PARITY           ;WRITE NORMAL
007710 012737 007750 000010     MOV     #T,@#RESVEC             ;RESERVED INSTRUCTION TIMEOUT
                                                                ;VECTOR ADDRESS
007716 012737 007750 001332     MOV     #T,@#$GDDAT             ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
007724 052777 000004 171670     BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
007732 012737 000340 000012     MOV     #340,@#RESVEC+2         ;NEW 'PS' FOR ERROR TRAP
007740 042777 000004 171654     BIC     #BIT2,@PARITY           ;WRITE NORMAL
007746 007000                   7000                            ;NON-RECOGNIZABLE OP-CODE
                                                                ;SHOULD ATTEMPT TO TRAP TO 'T:'
007750 104001           T:      HLT     +1                      ;DIDN'T ABORT
007752 000406                   BR      .+16                    ;GO TO RESET THE STACK
007754 042777 000001 171640 TO: BIC     #BIT0,@PARITY           ;DISABLE PARITY
007762 004037 011550            JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
007766 104003                   HLT     +3                      ;ABORTED INCORRECTLY
007770 012706 001100            MOV     #STACK,SP               ;RESET THE STACK
```

```
;*****************************************************************
;*****************************************************************
;*****************************************************************
;*****************************************************************
;
;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!!!
                ;
                ;*****************************************************************
                ;*****************************************************************
                ;*****************************************************************
                ;*****************************************************************
 2976
 2977
 2978                                   ;*****************************************************************
 2979                                   ;TEST 55        TEST NEW 'PC' FETCH
 2980                                   ;*****************************************************************
 2981  007774  000004                  TST55:  SCOPE
 2982                                   ;                       11/45 **** ROM STATE 355 ****
 2983                                   ;
 2984                                   ;       11/40 **** ROM STATE 333 ****
 2985  007776  004015                          JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
 2986  010000  010056                          V0                      ;ROUTINE ADDRESS
 2987  010002  012737  010052  000010          MOV     #V,@#RESVEC     ;RESERVED INSTRUCTION TIMEOUT
 2988                                                                  ;VECTOR ADDRESS
 2989  010010  012737  000340  000012          MOV     #340,@#RESVEC+2 ;NEW 'PS' FOR ERROR TRAP
 2990  010016  005737  001642                  TST     @#CPU40         ;ARE WE ON AN 11/40?
 2991  010022  001404                          BEQ     1$              ;BRANCH IF NO
 2992  010024  012737  010052  001332          MOV     #V,@#$GDDAT     ;STORE THIS VALUE OF THE PC THAT
 2993                                                                  ;SHOULD BE PUSHED ON THE STACK
 2994                                                                  ;IF A PARITY ABORT OCCURS
 2995                                                                  ;SINCE THE PC IS UPDATED FIRST
 2996                                                                  ;THEN THE VECTOR DATA TAKEN
 2997  010032  000403                          BR      2$              ;CONTINUE WITH TEST
 2998  010034  012737  000010  001332  1$:     MOV     #10,@#$GDDAT    ;STORE THE PC THAT SHOULD
 2999                                                                  ;BE PUSHED ON THE STACK
 3000                                                                  ;IF A PARITY ABORT OCCURS
 3001  010042  042777  000004  171552  2$:     BIC     #BIT2,@PARITY   ;WRITE NORMAL
 3002  010050  007000                          7000                    ;NON-RECOGNIZABLE OP-CODE
 3003                                                                  ;SHOULD ATTEMPT TO TRAP TO 'V:'
 3004  010052  104001                  V:      HLT     +1              ;DIDN'T ABORT
 3005  010054  000406                          BR      .+16            ;GO TO RESET THE STACK
 3006  010056  042777  000001  171536  V0:     BIC     #BIT0,@PARITY   ;DISABLE PARITY
 3007  010064  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
 3008  010070  104003                          HLT     +3              ;ABORTED INCORRECTLY
 3009  010072  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK
 3010
 3011
 3012
                ;*****************************************************************
```

```
;*********************************************************************
;*********************************************************************
;*********************************************************************
;
;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!
;
;*********************************************************************
;
;*********************************************************************
;*********************************************************************
;*********************************************************************
;*********************************************************************
3046
3047
3048                                    ;*****************************************************************
3049                                    ;TEST 56        TEST ABORT IN 'YELLOW' ZONE
3050                                    ;*****************************************************************
3051  010076  000004                    TST56:  SCOPE
3052                                    ;                       11/45 **** ROM STATE 177 ****
3053                                    ;
3054                                    ;       11/40 **** ROM STATE 267 ****
3055  010100  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
3056  010102  010162                            V1                              ;ROUTINE ADDRESS
3057  010104  005037  000340                    CLR     @#340                   ;CLEAR BOTTOM LIMIT LOCATION
3058                                                                            ;OF 'YELLOW' ZONE USING 'OTHER'
3059                                                                            ;PARITY
3060  010110  042777  000004  171504           BIC     #BIT2,@PARITY           ;WRITE NORMAL
3061  010116  012706  000376                    MOV     #376,SP                 ;SET STACK IN 'YELLOW' AREA
3062  010122  013727  000004                    MOV     @#4,(PC)+               ;SAVE CONTENTS OF LOC. 4 IN
3063                                                                            ;NEXT LOCATION
3064  010126  000000                    SAVLOC4:        .WORD   0               ;ORIGINAL CONTENTS OF LOC. 4
3065                                                                            ;GO HERE
3066  010130  012737  010166  000004            MOV     #V2,@#ERRVEC            ;SET UP A TIMEOUT VECTOR SERVICE
3067                                                                            ;ROUTINE ADDRESS FOR STACK VIO-
3068                                                                            ;LATION
3069  010136  012737  000340  000006            MOV     #340,@#ERRVEC+2         ;NEW PS ON TIMEOUT TRAP
```

```
3070  010144  012737  010156  001332          MOV     #.+12,@#$GDDAT  ;STORE THE PC THAT SHOULD
3071                                                                  ;BE PUSHED ON THE STACK
3072                                                                  ;IF A PARITY ABORT OCCURS
3073  010152  005766  177742                  TST     -36(SP)         ;REFERENCE BOTTOM LIMIT OF
3074                                                                  ;'YELLOW' ZONE USING REGISTER 6
3075                                                                  ;THIS INSTRUCTION SHOULD CAUSE
3076                                                                  ;AN ABORT 1ST, THEN A STACK
3077                                                                  ;VIOLATION
3078  010156  104010                          HLT     +10             ;DIDN'T ABORT OR RECOGNIZE THE
3079                                                                  ;STACK VIOLATION
3080  010160  000421                          BR      YELLOW          ;GO TO RESET STACK AND RESTORE
3081                                                                  ;TIMEOUT VECTORS
3082  010162  104011                  V1:     HLT     +11             ;ABORTED BUT STACK VIOLATION
3083                                                                  ;NOT RECOGNIZED
3084  010164  000417                          BR      YELLOW          ;GO TO RESET STACK AND RESTORE
3085                                                                  ;TIMEOUT VECTORS
3086  010166  042777  000001  171426  V2:     BIC     #BIT0,@PARITY   ;DISABLE PARITY
3087  010174  005737  000372                  TST     @#372           ;STACK VIOLATION PICKED UP -
3088                                                                  ;WAS THE PARITY ABORT?
3089  010200  001002                          BNE     1$              ;BRANCH IF YES
3090  010202  104012                          HLT     +12             ;STACK VIOLATION PICKED UP
3091                                                                  ;BUT ABORT NOT RECOGNIZED
3092  010204  000407                          BR      YELLOW          ;GO TO RESET STACK AND RESTORE
3093                                                                  ;TIMEOUT VECTORS
3094  010206  012706  001100          1$:     MOV     #STACK,SP       ;RESET STACK BACK TO NORMAL
3095  010212  013746  000372                  MOV     @#372,-(SP)     ;PUSH ABORT PC ONTO KERNAL
3096                                                                  ;STACK OUT OF VIOLATION AREAS
3097                                                                  ;FOR CHECKING PURPOSES
3098  010216  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
3099  010222  104003                          HLT     +3              ;ABORTED INCORRECTLY
3100  010224  012706  001100          YELLOW: MOV     #STACK,SP       ;RESET STACK BACK TO NORMAL
3101  010230  013737  010126  000004          MOV     @#SAVLOC4,@#4   ;RESTORE CONTENTS OF LOC. 4
3102  010236  005037  000006                  CLR     @#6             ;RESTORE CONTENTS OF LOC. 6
3103  010242  005037  000372                  CLR     @#372           ;RESET TRAPCATCHER LOCATION
3104                                                                  ;FOR NEXT TEST
3105  010246  005037  000340                  CLR     @#340           ;CLEAR BOTTOM LIMIT LOCATION
3106                                                                  ;OF 'YELLOW' ZONE USING NORMAL
3107                                                                  ;PARITY
3108
3109
3110

        ;*****************************************************************************
        ;*****************************************************************************
        ;*****************************************************************************
        ;*****************************************************************************
        ;
        ;
        ;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.
        ;
        ;
        ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************
        ;
3150
3151
3152                                    ;*****************************************************************
3153                                    ;TEST 57        TEST ABORT IN 'RED' ZONE
3154                                    ;*****************************************************************
3155    010252  000004                  TST57:  SCOPE
3156                                    ;                       11/45 **** ROM STATE 177 ****
3157                                    ;
3158                                    ;       11/40 **** ROM STATE 267 ****
3159    010254  004015                          JSR     R0,(R5)                 ;SET UP A PARITY VECTOR SERVICE
3160    010256  010336                          V3                              ;ROUTINE ADDRESS
3161    010260  005037  000336                  CLR     @#336                   ;CLEAR 1ST LOCATION OF 'RED'
3162                                                                            ;ZONE USING 'OTHER' PARITY
3163    010264  042777  000004  171330          BIC     #BIT2,@PARITY           ;WRITE NORMAL
3164    010272  012706  000376                  MOV     #376,SP                 ;SET STACK IN 'YELLOW'AREA
3165    010276  013727  000004                  MOV     @#4,(PC)+               ;SAVE CONTENTS OF LOC. 4
3166                                                                            ;IN NEXT LOCATION
3167    010302  000000          HOLDLOC4:       .WORD   0                       ;ORIGINAL CONTENTS OF LOC. 4
3168                                                                            ;GO HERE
3169    010304  012737  010342  000004          MOV     #V4,@#ERRVEC            ;SET UP A TIMEOUT VECTOR SERVICE
3170                                                                            ;ROUTINE ADDRESS FOR STACK
3171                                                                            ;VIOLATION
3172    010312  012737  000340  000006          MOV     #340,@#ERRVEC+2         ;NEW PS ON TIMEOUT TRAP
3173    010320  012737  010332  001332          MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
3174                                                                            ;BE PUSHED ON THE STACK
3175                                                                            ;IF A PARITY ABORT OCCURS
3176    010326  005766  177740                  TST     -40(SP)                 ;REFERENCE 1ST LOCATION OF 'RED'
3177                                                                            ;ZONE USING REGISTER 6
3178                                                                            ;THIS INSTRUCTION SHOULD CAUSE
3179                                                                            ;AN ABORT 1ST, THEN A STACK
3180                                                                            ;VIOLATION
3181    010332  104010                          HLT     +10                     ;DIDN'T ABORT OR RECOGNIZE THE
3182                                                                            ;STACK VIOLATION
3183    010334  000421                          BR      RED                     ;GO TO RESET STACK AND RESTORE
3184                                                                            ;TIMEOUT VECTORS
3185    010336  104011          V3:             HLT     +11                     ;ABORTED BUT STACK VIOLATION
3186                                                                            ;NOT RECOGNIZED
3187    010340  000417                          BR      RED                     ;GO TO RESET STACK AND RESTORE
3188                                                                            ;TIMEOUT VECTORS
3189    010342  042777  000001  171252  V4:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
3190    010350  005737  000372                  TST     @#372                   ;STACK VIOLATION PICKED UP -
3191                                                                            ;WAS THE PARITY ABORT?
3192    010354  001002                          BNE     1$                      ;BRANCH IF YES
```

```
3193  010356  104012                    HLT     +12                     ;STACK VIOLATION PICKED UP BUT
3194                                                                      ;ABORT NOT RECOGNIZED
3195  010360  000407                    BR      RED                     ;GO TO RESET STACK AND RESTORE
3196                                                                      ;TIMEOUT VECTORS
3197  010362  012706  001100      1$:   MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
3198  010366  013746  000372            MOV     @#372,-(SP)             ;PUSH ABORT PC ONTC KERNAL
3199                                                                      ;STACK OUT OF VIOLATION AREAS
3200                                                                      ;FOR CHECKING PURPOSES
3201  010372  004037  011550            JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
3202  010376  104003                    HLT     +3                      ;ABORTED INCORRECTLY
3203  010400  012706  001100      RED:  MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
3204  010404  013737  010302  000004    MOV     @#HOLDLOC4,@#4          ;RESTORE CONTENTS OF LOC. 4
3205  010412  005037  000006            CLR     @#6                     ;RESTORE CONTENTS OF LOC. 6
3206  010416  005037  000372            CLR     @#372                   ;RESET TRAPCATCHER LOCATION
3207                                                                      ;FOR NEXT TEST
3208  010422  005037  000336            CLR     @#336                   ;CLEAR 1ST LOCATION OF 'RED'
3209                                                                      ;ZONE USING NORMAL PARITY
3210                              ;*********************************************************************
3211                              ;TEST 60        TEST (INDEX WORD)  SM0,DM7  MOV INSTRUCTION
3212                              ;*********************************************************************
3213  010426  000004              TST60:  SCOPE
3214                              ;                       11/45 **** ROM STATE 7 ****
3215                              ;
3216                              ;       11/40 **** ROM STATE 206 ****
3217  010430  005037  001624            CLR     @#PSPCORZONES           ;CLEAR PS, PC OR ZONES
3218                                                                      ;ABORT FLAG
3219  010434  004015                    JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
3220  010436  010514                    W                               ;ROUTINE ADDRESS
3221  010440  042777  000004  171154    BIC     #BIT2,@PARITY           ;WRITE NORMAL
3222  010446  010100                    MOV     R1,R0                   ;SET UP FOR A DATO
3223  010450  012720  010070            MOV     #10070,(R0)+            ;MOVE THE INSTRUCTION
3224                                                                      ;'MOV R0,@-2(R0)' TO PARITY
3225                                                                      ;MEMORY AREA
3226  010454  052777  000004  171140    BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
3227  010462  012720  177776            MOV     #-2,(R0)+               ;WRITE THE INDEX WORD IN NEXT
3228                                                                      ;PARITY MEMORY AREA LOCATION
3229  010466  042777  000004  171126    BIC     #BIT2,@PARITY           ;WRITE NORMAL
3230  010474  010037  001332            MOV     R0,@#$GDDAT             ;STORE THE PC THAT SHOULD
3231                                                                      ;BE PUSHED ON THE STACK
3232                                                                      ;IF A PARITY ABORT OCCURS
3233  010500  012710  000203            MOV     #203,(R0)               ;MOVE 'RTS R3' INTO NEXT
3234                                                                      ;PARITY MEMORY AREA LOCATION
3235                                                                      ;IN CASE WE DON'T ABORT
3236  010504  004360  177774            JSR     R3,-4(R0)               ;GO TO PARITY MEMORY AREA
3237  010510  104001                    HLT     +1                      ;DIDN'T ABORT
3238  010512  000410                    BR      .+22                    ;GO TO NEXT TEST
3239  010514  042777  000001  171100  W:BIC     #BIT0,@PARITY           ;DISABLE PARITY
3240  010522  004037  011550            JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
3241  010526  104003                    HLT     +3                      ;ABORTED INCORRECTLY
3242  010530  012706  001100            MOV     #STACK,SP               ;RESET THE STACK
3243                              ;*********************************************************************
3244                              ;TEST 61        TEST (INDEX WORD)  SM6,DM0  CMP INSTRUCTION
3245                              ;*********************************************************************
3246  010534  000004              TST61:  SCOPE
3247                              ;                       11/45 **** ROM STATE 26 ****
3248                              ;
```

```
3249                                    ;       11/40 **** ROM STATE 241 ****
3250  010536  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
3251  010540  010616                            XX                      ;ROUTINE ADDRESS
3252  010542  042777  000004  171052            BIC     #BIT2,@PARITY   ;WRITE NORMAL
3253  010550  010100                            MOV     R1,R0           ;SET UP FOR A DATO
3254  010552  012720  026000                    MOV     #26000,(R0)+    ;MOVE THE INSTRUCTION
3255                                                                    ;'CMP -2(R0),R0' TO PARITY
3256                                                                    ;MEMORY AREA
3257  010556  052777  000004  171036            BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
3258  010564  012720  177776                    MOV     #-2,(R0)+       ;WRITE THE INDEX WORD IN NEXT
3259                                                                    ;PARITY MEMORY AREA LOCATION
3260  010570  042777  000004  171024            BIC     #BIT2,@PARITY   ;WRITE NORMAL
3261  010576  010037  001332                    MOV     R0,@#$GDDAT     ;STORE THE PC THAT SHOULD
3262                                                                    ;BE PUSHED ON THE STACK
3263                                                                    ;IF A PARITY ABORT OCCURS
3264  010602  012710  000203                    MOV     #203,(R0)       ;MOVE 'RTS R3' INTO NEXT
3265                                                                    ;PARITY MEMORY AREA LOCATION
3266                                                                    ;IN CASE WE DON'T ABORT
3267  010606  004360  177774                    JSR     R3,-4(R0)       ;GO TO PARITY MEMORY AREA
3268  010612  104001                            HLT     +1              ;DIDN'T ABORT
3269  010614  000410                            BR      .+22            ;GO TO NEXT TEST
3270  010616  042777  000001  170776  XX:       BIC     #BIT0,@PARITY   ;DISABLE PARITY
3271  010624  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
3272  010630  104003                            HLT     +3              ;ABORTED INCORRECTLY
3273  010632  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
3274                                  ;*****************************************************************
3275                                  ;TEST 62        TEST (INDEX WORD)  SMD,DM6  MOV INSTRUCTION
3276                                  ;*****************************************************************
3277  010636  000004                  TST62:  SCOPE
3278                                  ;                       11/45 **** ROM STATE 6 ****
3279                                  ;
3280                                  ;       11/40 **** ROM STATE 206 ****
3281  010640  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
3282  010642  010720                            Y                       ;ROUTINE ADDRESS
3283  010644  042777  000004  170750            BIC     #BIT2,@PARITY   ;WRITE NORMAL
3284  010652  010100                            MOV     R1,R0           ;SET UP FOR A DATO
3285  010654  012720  010060                    MOV     #10060,(R0)+    ;MOVE THE INSTRUCTION
3286                                                                    ;'MOV R0,-2(R0)' TO PARITY
3287                                                                    ;MEMORY AREA
3288  010660  052777  000004  170734            BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
3289  010666  012720  177776                    MOV     #-2,(R0)+       ;WRITE THE INDEX WORD IN NEXT
3290                                                                    ;PARITY MEMORY AREA LOCATION
3291  010672  042777  000004  170722            BIC     #BIT2,@PARITY   ;WRITE NORMAL
3292  010700  010037  001332                    MOV     R0,@#$GDDAT     ;STORE THE PC THAT SHOULD
3293                                                                    ;BE PUSHED ON THE STACK
3294                                                                    ;IF A PARITY ABORT OCCURS
3295  010704  012710  000203                    MOV     #203,(R0)       ;MOVE 'RTS R3' INTO NEXT
3296                                                                    ;PARITY MEMORY AREA LOCATION
3297                                                                    ;IN CASE WE DON'T ABORT
3298  010710  004360  177774                    JSR     R3,-4(R0)       ;GO TO PARITY MEMORY AREA
3299  010714  104001                            HLT     +1              ;DIDN'T ABORT
3300  010716  000410                            BR      .+22            ;GO TO NEXT TEST
3301  010720  042777  000001  170674  Y:        BIC     #BIT0,@PARITY   ;DISABLE PARITY
3302  010726  004037  011550                    JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
3303  010732  104003                            HLT     +3              ;ABORTED INCORRECTLY
3304  010734  012706  001100                    MOV     #STACK,SP       ;RESET THE STACK
```

```
3305                                    ;*****************************************************************
3306                                    ;TEST 63        TEST   BPL INSTRUCTION (-BCOK)
3307                                    ;*****************************************************************
3308  010740  000004                    TST63:  SCOPE
3309                                    ;                       11/45 **** ROM STATE 321 ****
3310                                    ;
3311                                    ;       11/40 **** ROM STATE 1 ****
3312  010742  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
3313  010744  011056                            ZO                              ;ROUTINE ADDRESS
3314  010746  042777  000004  170646            BIC     #BIT2,@PARITY           ;WRITE NORMAL
3315  010754  010102                            MOV     R1,R2                   ;SET UP FOR A DATO
3316  010756  162702  000004                    SUB     #4,R2                   ;CALCULATE THE START
3317                                                                            ;ADDRESS FOR THIS TEST
3318  010762  012722  012700                    MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
3319                                                                            ;'MOV #-1,R0' TO PARITY
3320                                                                            ;MEMORY AREA
3321  010766  012722  177777                    MOV     #-1,(R2)+               ;MOVE A -1 INTO NEXT
3322                                                                            ;PARITY MEMORY AREA LOCATION
3323  010772  012722  100001                    MOV     #100001,(R2)+           ;MOVE THE INSTRUCTION
3324                                                                            ;'BPL .+4' INTO NEXT PARITY
3325                                                                            ;MEMORY AREA LOCATION
3326  010776  052777  000004  170616            BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
3327  011004  005737  001642                    TST     @#CPU40                 ;ARE WE ON AN 11/40?
3328  011010  001405                            BEQ     2$                      ;BRANCH IF NO
3329                                                                            ;SINCE THE PC WILL BE UPDATED
3330                                                                            ;ON THE PARITY ABORT
3331  011012  010237  001332                    MOV     R2,@#$GDDAT             ;STORE THIS PC THAT SHOULD
3332                                                                            ;BE PUSHED ON THE STACK
3333                                                                            ;IF A PARITY ABORT OCCURS
3334                                                                            ;SINCE THE PC IS NOT UPDATED
3335                                                                            ;ON THE PARITY ABORT
3336  011016  012722  000240                    MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
3337                                                                            ;PARITY MEMORY AREA LOCATION
3338  011022  000404                            BR      1$                      ;CONTINUE WITH TEST
3339  011024  012722  000240            2$:     MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
3340                                                                            ;PARITY MEMORY AREA LOCATION
3341  011030  010237  001332                    MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
3342                                                                            ;BE PUSHED ON THE STACK
3343                                                                            ;IF A PARITY ABORT OCCURS
3344  011034  042777  000004  170560    1$:     BIC     #BIT2,@PARITY           ;WRITE NORMAL
3345  011042  012712  000203                    MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
3346                                                                            ;PARITY MEMORY AREA LOCATION
3347                                                                            ;IN CASE WE DON'T ABORT
3348  011046  004362  177770                    JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
3349  011052  104001                            HLT     +1                      ;DIDN'T ABORT
3350  011054  000410                            BR      .+22                    ;GO TO NEXT TEST
3351  011056  042777  000001  170536    ZO:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
3352  011064  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
3353  011070  104003                            HLT     +3                      ;ABORTED INCORRECTLY
3354  011072  012706  001100                    MOV     #STACK,SP               ;RESET THE STACK
3355                                    ;*****************************************************************
3356                                    ;TEST 64        TEST   BNE INSTRUCTION   (-BCOK)
3357                                    ;*****************************************************************
3358  011076  000004                    TST64:  SCOPE
3359                                    ;                       11/45 **** ROM STATE 322 ****
3360                                    ;
```

```
3361                                        ;       11/40 **** ROM STATE 1 ****
3362  011100  004015                              JSR     R0,(R5)             ;SET UP PARITY VECTOR SERVICE
3363  011102  011214                              Z1                          ;ROUTINE ADDRESS
3364  011104  042777  000004  170510              BIC     #BIT2,@PARITY       ;WRITE NORMAL
3365  011112  010102                              MOV     R1,R2               ;SET UP FOR A DATO
3366  011114  162702  000004                      SUB     #4,R2               ;CALCULATE THE START ADDRESS
3367                                                                          ;FOR THIS TEST
3368  011120  012722  012700                      MOV     #12700,(R2)+        ;MOVE THE INSTRUCTION
3369                                                                          ;'MOV  #0,R0' INTO PARITY
3370                                                                          ;MEMORY AREA
3371  011124  012722  000000                      MOV     #0,(R2)+            ;MOVE A 0 INTO NEXT PARITY
3372                                                                          ;MEMORY AREA LOCATION
3373  011130  012722  001001                      MOV     #1001,(R2)+         ;MOVE THE INSTRUCTION
3374                                                                          ;'BNE .+4' INTO NEXT
3375                                                                          ;PARITY MEMORY AREA LOCATION
3376  011134  052777  000004  170460              BIS     #BIT2,@PARITY       ;WRITE OTHER PARITY
3377  011142  005737  001642                      TST     @#CPU40             ;ARE WE ON AN 11/40?
3378  011146  001405                              BEQ     2$                  ;BRANCH IF NO
3379                                                                          ;SINCE THE PC WILL BE UPDATED
3380                                                                          ;ON THE PARITY ABORT
3381  011150  010237  001332                      MOV     R2,@#$GDDAT         ;STORE THIS PC THAT SHOULD
3382                                                                          ;BE PUSHED ON THE STACK
3383                                                                          ;IF A PARITY ABORT OCCURS
3384                                                                          ;SINCE THE PC WILL NOT BE
3385                                                                          ;UPDATED ON THE PARITY ABORT
3386  011154  012722  000240                      MOV     #240,(R2)+          ;MOVE A 'NOP' INTO NEXT
3387                                                                          ;PARITY MEMORY AREA LOCATION
3388  011160  000404                              BR      1$                  ;CONTINUE WITH TEST
3389  011162  012722  000240          2$:         MOV     #240,(R2)+          ;MOVE A 'NOP' INTO NEXT
3390                                                                          ;PARITY MEMORY AREA LOCATION
3391  011166  010237  001332                      MOV     R2,@#$GDDAT         ;STORE THE PC THAT SHOULD
3392                                                                          ;BE PUSHED ON THE STACK
3393                                                                          ;IF A PARITY ABORT OCCURS
3394  011172  042777  000004  170422  1$:         BIC     #BIT2,@PARITY       ;WRITE NORMAL
3395  011200  012712  000203                      MOV     #203,(R2)           ;MOVE 'RTS R3' INTO NEXT
3396                                                                          ;PARITY MEMORY AREA LOCATION
3397                                                                          ;IN CASE WE DON'T ABORT
3398  011204  004362  177770                      JSR     R3,-10(R2)          ;GO TO PARITY MEMORY AREA
3399  011210  104001                              HLT     +1                  ;DIDN'T ABORT
3400  011212  000410                              BR      .+22                ;GO TO NEXT TEST
3401  011214  042777  000001  170400  Z1:         BIC     #BIT0,@PARITY       ;DISABLE PARITY
3402  011222  004037  011550                      JSR     R0,@#CHECKLOC       ;CHECK FOR GOOD ABORT
3403  011226  104003                              HLT     +3                  ;ABORTED INCORRECTLY
3404  011230  012706  001100                      MOV     #STACK,SP           ;RESET THE STACK
3405                                        ;*****************************************************************
3406                                        ;TEST 65        TEST   BEQ INSTRUCTION   (-BCOK)
3407                                        ;*****************************************************************
3408  011234  000004                        TST65:  SCOPE
3409                                        ;                       11/45 **** ROM STATE 324 ****
3410                                        ;
3411                                        ;       11/40 **** ROM STATE 1 ****
3412  011236  004015                              JSR     R0,(R5)             ;SET UP PARITY VECTOR SERVICE
3413  011240  011352                              Z2                          ;ROUTINE ADDRESS
3414  011242  042777  000004  170352              BIC     #BIT2,@PARITY       ;WRITE NORMAL
3415  011250  010102                              MOV     R1,R2               ;SET UP FOR A DATO
3416  011252  162702  000004                      SUB     #4,R2               ;CALCULATE THE START ADDRESS
```

```
3417
3418  011256  012722  012700                  MOV     #12700,(R2)+    ;FOR THIS TEST
3419                                                                    ;MOVE THE INSTRUCTION
3420                                                                    ;'MOV #-1,R0' TO PARITY MEMORY
3421  011262  012722  177777                  MOV     #-1,(R2)+       ;AREA
3422                                                                    ;MOVE A -1 INTO NEXT PARITY
3423  011266  012722  001401                  MOV     #1401,(R2)+     ;MEMORY AREA LOCATION
3424                                                                    ;MOVE THE INSTRUCTION
3425                                                                    ;'BEQ .+4' INTO NEXT PARITY
3426  011272  052777  000004  170322          BIS     #BIT2,@PARITY   ;MEMORY AREA LOCATION
3427  011300  005737  001642                  TST     @#CPU40         ;WRITE OTHER PARITY
3428  011304  001405                          BEQ     2$              ;ARE WE ON AN 11/40?
3429                                                                    ;BRANCH IF NO
3430                                                                    ;SINCE THE PC WILL BE UPDATED
3431  011306  010237  001332                  MOV     R2,@#$GDDAT     ;ON THE PARITY ABORT
3432                                                                    ;STORE THIS PC THAT SHOULD
3433                                                                    ;BE PUSHED ON THE STACK
3434                                                                    ;IF A PARITY ABORT OCCURS
3435                                                                    ;SINCE THE PC WILL NOT BE
3436  011312  012722  000240                  MOV     #240,(R2)+      ;UPDATED ON THE PARITY ABORT
3437                                                                    ;MOVE A 'NOP' INTO NEXT
3438  011316  000404                          BR      1$              ;PARITY MEMORY AREA LOCATION
3439  011320  012722  000240          2$:     MOV     #240,(R2)+      ;CONTINUE WITH TEST
3440                                                                    ;MOVE A 'NOP' INTO NEXT
3441  011324  010237  001332                  MOV     R2,@#$GDDAT     ;PARITY MEMORY AREA LOCATION
3442                                                                    ;STORE THE PC THAT SHOULD
3443                                                                    ;BE PUSHED ON THE STACK
3444  011330  042777  000004  170264  1$:     BIC     #BIT2,@PARITY   ;IF A PARITY ABORT OCCURS
3445  011336  012712  000203                  MOV     #203,(R2)       ;WRITE NORMAL
3446                                                                    ;MOVE 'RTS R3' INTO NEXT
3447                                                                    ;PARITY MEMORY AREA LOCATION
3448  011342  004362  177770                  JSR     R3,-10(R2)      ;IN CASE WE DON'T ABORT
3449  011346  104001                          HLT     +1              ;GO TO PARITY MEMORY AREA
3450  011350  000410                          BR      .+22            ;DIDN'T ABORT
3451  011352  042777  000001  170242  Z2:     BIC     #BIT0,@PARITY   ;GO TO NEXT TEST
3452  011360  004037  011550                  JSR     R0,@#CHECKLOC   ;DISABLE PARITY
3453  011364  104003                          HLT     +3              ;CHECK FOR GOOD ABORT
3454  011366  012706  001100                  MOV     #STACK,SP       ;ABORTED INCORRECTLY
3455                                  ;*****************************************************************
3456                                  ;TEST 66        END OF PROGRAM
3457                                  ;*****************************************************************
3458  011372  000004                  TST66:  SCOPE
3459  011374  042777  100005  170220          BIC     #BIT15!BIT2!BIT0,@PARITY        ;DISABLE ALL PARITY
3460                                                                    ;AND CLEAR ERROR BIT!
3461  011402  005737  001630                  TST     @#USERTYPE      ;DID THE USER SELECT THE REGISTER?
3462  011406  001014                          BNE     1$              ;BRANCH IF SO AND DON'T STEP
3463                                                                    ;UP THE TABLE
3464  011410  062737  000002  001336          ADD     #2,@#$REGAD     ;STEP UP TO NEXT REGISTER
3465  011416  062737  000002  001414          ADD     #2,@#$TMPAD     ;STEP UP TO CORRESPONDING
3466                                                                    ;PARITY MEMORY
3467  011424  062737  000002  001416          ADD     #2,@#$SETAD     ;STEP UP TO NEXT OFFSET - THIS
3468                                                                    ;IS ONLY APPLICABLE IF MEMORY
3469                                                                    ;MGMT IS TURNED ON
3470  011432  062737  000002  001446          ADD     #2,@#NTERAD     ;STEP UP TO NEXT INTERLEAVE
3471                                                                    ;VALUE
3472  011440  004337  011474          1$:     JSR     R3,@#FLAGSCLR   ;CLEAR PERTINENT FLAGS
```

```
3473  011444  000167  000320                  JMP     $EOP                    ;GO TO RING-A-DING
3474                                                                          ;BEFORE REITERATING THE PROGRAM
3475
3476
3477
3478                                  ;*************************************************************************
3479                                  ;
3480                                  ;ROUTINE FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS
3481                                  ;
3482                                  ;*************************************************************************
3483  011450  012077  170144         VECSET: MOV     (R0)+,@INTVEC           ;WRITE ADDRESS INTO LOCATION 114
3484  011454  052777  000005  170140         BIS     #BIT2!BIT0,@PARITY      ;WRITE OTHER PARITY AND DISABLE
3485  011462  000200                         RTS     R0                      ;RETURN TO TESTING
3486                                  ;*************************************************************
3487                                  ;
3488                                  ;ROUTINE TO RESET AND GO BACK TO TABLE BEGINNING
3489                                  ;
3490                                  ;*************************************************************************
3491  011464  004337  011506         RESTART:        JSR     R3,@#INITIALIZE
3492  011470  000167  000274                 JMP     $EOP                    ;GO TO RING-A-DING
3493                                  ;*************************************************************
3494                                  ;
3495                                  ;ROUTINE TO CLEAR PERTINENT FLAGS BEFORE PASSING THRU THE PROGRAM
3496                                  ;WITH ANOTHER TABLE ENTRY
3497                                  ;
3498                                  ;*************************************************************************
3499  011474  005037  001624         FLAGSCLR:       CLR     @#PSPCORZONES           ;CLEAR PS, PC AND ZONES ABORT
3500                                                                          ;FLAG
3501  011500  005037  001626                 CLR     @#MSREGFLAG             ;CLEAR MS11 REGISTER PRESENCE
3502                                                                          ;FLAG
3503  011504  000203                         RTS     R3                      ;RETURN
3504                                  ;*************************************************************
3505                                  ;
3506                                  ;ROUTINE TO COMPLETELY REINITIALIZE BEFORE RESTARTING PROGRAM OVER
3507                                  ;AT THE BEGINNING OF THE TABLE
3508                                  ;
3509                                  ;*************************************************************************
3510  011506  005037  001624         INITIALIZE:     CLR     @#PSPCORZONES           ;CLEAR PS, PC AND ZONES
3511                                                                          ;ABORT FLAG
3512  011512  005037  001626                 CLR     @#MSREGFLAG             ;CLEAR MS11 REGISTER PRESENCE
3513                                                                          ;FLAG
3514  011516  012737  001340  001336         MOV     #$REG0,@#$REGAD         ;MOVE FIRST REGISTER CONTAINER
3515                                                                          ;INTO $REGAD WHICH IS USED TO
3516                                                                          ;POINT TO THE PARITY REGISTER
3517                                                                          ;TABLE
3518  011524  012737  001366  001414         MOV     #$TMP0,@#$TMPAD         ;MOVE FIRST MEMORY CONTAINER
3519                                                                          ;INTO $TMPAD WHICH IS USED TO
3520                                                                          ;POINT TO THE MEMORY LOCATION
3521                                                                          ;TABLE
3522  011532  012737  001420  001416         MOV     #$SET0,@#$SETAD         ;MOVE FIRST OFFSET CONTAINER
3523                                                                          ;INTO $SETAD WHICH IS USED TO
3524                                                                          ;POINT TO THE OFFSET LOCATION
3525                                                                          ;TABLE IF MEMORY MGMT IS USED
3526  011540  012737  001450  001446         MOV     #NTER0,@#NTERAD         ;MOVE 1ST INTERLEAVE CONTAINER
3527                                                                          ;INTO 'INTERAD' WHICH IS USED
3528                                                                          ;TO POINT TO THE INTERLEAVE
```

```
3529
3530  011546  000203                          RTS     R3                      ;LOCATION TABLE
3531                                                                          ;RETURN
3532
3533                                  ;*************************************************************************
3534                                  ;
3535                                  ;SUBROUTINE TO CHECK THAT IF A TEST HAS ABORTED IT DID INDEED
3536                                  ;ABORT IN THE PROPER PLACE.  IT IS QUITE CONCEIVABLE THAT A TEST
3537                                  ;THAT SHOULD HAVE ABORTED IN THE TWO LOCATION MAP AREA ABORTED
3538                                  ;ON THE FETCH OF THE INSTRUCTION THAT WAS TO CAUSE THE ABORT.
3539                                  ;THIS SUBROUTINE WILL FLAG SUCH OCCURRENCES.  WITHOUT THIS CHECK
3540                                  ;THE PROGRAM WOULD APPEAR TO HAVE RUN PROPERLY.
3541                                  ;
3542                                  ;BOTH THE CORRECT HIGH ORDER ERROR ADDRESS BITS AND PROPER PC
3543                                  ;PUSH ON THE STACK ARE LOOKED FOR AFTER A PARITY ABORT OCCURS.
3544                                  ;
3545                                  ;*************************************************************************
3546  011550  010246                  CHECKLOC:       MOV     R2,-(SP)        ;SAVE R2 CONTENTS ON STACK
3547  011552  010346                          MOV     R3,-(SP)                ;SAVE R3 CONTENTS ON STACK
3548  011554  010446                          MOV     R4,-(SP)                ;SAVE R4 CONTENTS ON STACK
3549  011556  005002                          CLR     R2                      ;CLEAR ERROR ADDRESS BIT COMPARE
3550  011560  005004                          CLR     R4                      ;REGISTERS IN CASE WE HAVE AN OLD
3551                                                                          ;MOS DESIGN THAT DOESN'T HAVE
3552                                                                          ;ADDRESS BITS
3553  011562  005737  001626                  TST     @#MSREGFLAG             ;IS AN MF11 OR MS11 PARITY
3554                                                                          ;OPTION BEING TESTED?
3555  011566  001027                          BNE     7$                      ;BRANCH IF MS11 AND DON'T DO
3556                                                                          ;ADDRESS BITS CHECKING
3557  011570  005737  001624                  TST     @#PSPCORZONES           ;ARE WE DOING A PS OR PC FETCH
3558                                                                          ;OR ZONE ABORT TEST?
3559  011574  001402                          BEQ     5$                      ;BRANCH IF NO
3560  011576  005002                          CLR     R2                      ;SET THE PARITY REGISTER HIGH
3561                                                                          ;ORDER ADDRESS BITS VALUE
3562                                                                          ;(BITS 5 THRU 11) TO ZERO FOR
3563                                                                          ;THE PS OR PC FETCH ABORT TESTS
3564  011600  000414                          BR      2$                      ;GO CHECK IT AGAINST ACTUAL
3565                                                                          ;PARITY REGISTER VALUE
3566  011602  005737  002304          5$:     TST     @#$KT11                 ;MEMORY MANAGEMENT ON?
3567  011606  100003                          BPL     8$                      ;BRANCH IF NO
3568  011610  017702  167602                  MOV     @$SETAD,R2              ;PICK UP THE OFFSET VALUE - IT
3569                                                                          ;SHOULD BE THE VALUE THAT WILL
3570                                                                          ;APPEAR IN THE PARITY REGISTER
3571                                                                          ;ERROR ADDRESS BITS
3572  011614  000406                          BR      2$                      ;GO TO CHECK!
3573  011616  012703  017400          8$:     MOV     #17400,R3               ;GET A FIRST POSSIBLE ABORT
3574                                                                          ;LOCATION AREA
3575  011622  012702  000140                  MOV     #140,R2                 ;GET THE FIRST POSSIBLE PARITY
3576                                                                          ;REGISTER HIGH ORDER ADDRESS
3577                                                                          ;BITS VALUE (BITS 5 THRU 11)
3578  011626  020103          1$:     CMP     R1,R3                   ;IS THIS THE ABORT AREA BEING
3579                                                                          ;USED?
3580  011630  101027                          BHI     3$                      ;NO - SEE IF IT'S THE NEXT ONE
3581  011632  017704  167764          2$:     MOV     @PARITY,R4              ;PROCEED TO SEE IF
3582                                                                          ;IT WAS A PROPER ABORT BY LOOKING
3583                                                                          ;AT THE HIGH ORDER ADDRESS BITS
3584                                                                          ;OF THE PARITY REGISTER
```

```
3585
3586  011636  042704  170037              BIC     #170037,R4      ;(BITS 5 THRU 11)
3587  011642  020402                      CMP     R4,R2           ;CLEAR ALL BITS EXCEPT 5 THRU 11
3588                                                              ;ARE THE ERROR ADDRESS BITS
3589  011644  001026                      BNE     4$              ;WHAT THEY SHOULD BE?
3590  011646  010237  001326      7$:     MOV     R2,@#$GDADR     ;BRANCH IF NO
3591                                                              ;STORE WHAT THE ADDRESS BITS
3592  011652  010437  001330              MOV     R4,@#$BDADR     ;SHOULD HAVE BEEN
3593                                                              ;STORE WHAT THE ADDRESS BITS
3594  011656  012604                      MOV     (SP)+,R4        ;WERE
3595  011660  012603                      MOV     (SP)+,R3        ;RESTORE R4 CONTENTS
3596  011662  012602                      MOV     (SP)+,R2        ;RESTORE R3 CONTENTS
3597                                                              ;RESTORE R2 CONTENTS
3598                      ;IF WE HAVE REACHED THIS PATH 1 OF 2 CONDITIONS EXIST --
3599                      ;WE HAVE AN OLD MOS DESIGN WITH NO ADDRESS BITS, OR WE HAVE
3600                      ;CORE OR THE NEW MOS DESIGN WITH ADDRESS BITS OK!!
3601
3602  011664  026637  000002  001332      CMP     2(SP),@#$GDDAT  ;WAS THE CORRECT PC PUSHED
3603                                                              ;ON THE STACK?
3604  011672  001404                      BEQ     9$              ;BRANCH IF YES
3605  011674  016637  000002  001334      MOV     2(SP),@#$BDDAT  ;SAVE INCORRECT PC FOR PRINTOUT
3606  011702  000200                      RTS     R0              ;GO BACK TO INDICATE BAD ABORT
3607  011704  005720              9$:     TST     (R0)+           ;STEP UP RETURN ADDRESS
3608                                                              ;TO BYPASS THE ERROR HLT
3609  011706  000200                      RTS     R0              ;RETURN TO CONTINUE TESTING
3610  011710  062703  004000      3$:     ADD     #4000,R3        ;STEP UP TO THE NEXT POSSIBLE
3611                                                              ;ABORT LOCATION AREA
3612  011714  062702  000040              ADD     #40,R2          ;CHANGE THE VALUE OF THE HIGH
3613                                                              ;ORDER ADDRESS BITS VALUE TO BE
3614                                                              ;CHECKED
3615  011720  000742                      BR      1$              ;GO BACK TO CHECK THIS ONE
3616  011722  010237  001326      4$:     MOV     R2,@#$GDADR     ;STORE WHAT THE ADDRESS BITS
3617                                                              ;SHOULD HAVE BEEN
3618  011726  010437  001330              MOV     R4,@#$BDADR     ;STORE WHAT THE
3619                                                              ;ADDRESS BITS WERE
3620  011732  012604                      MOV     (SP)+,R4        ;RESTORE R4 CONTENTS
3621  011734  012603                      MOV     (SP)+,R3        ;RESTORE R3 CONTENTS
3622  011736  012602                      MOV     (SP)+,R2        ;RESTORE R2 CONTENTS
3623
3624                      ;IF WE HAVE REACHED THIS PATH WE HAVE CORE OR THE NEW MOS DESIGN
3625                      ;WITH BAD ADDRESS BITS WHICH SHOULD INDICATE AN IMPROPER ABORT!!
3626
3627  011740  026637  000002  001332      CMP     2(SP),@#$GDDAT  ;WAS THE CORRECT PC PUSHED
3628                                                              ;ON THE STACK?
3629  011746  001404                      BEQ     10$             ;BRANCH IF YES
3630  011750  016637  000002  001334      MOV     2(SP),@#$BDDAT  ;SAVE INCORRECT PC FOR PRINTOUT
3631  011756  000200                      RTS     R0              ;GO BACK TO INDICATE BAD ABORT
3632  011760  016737  167346  001334 10$: MOV     $GDDAT,@#$BDDAT ;RECORD THE FACT THAT THE
3633                                                              ;CORRECT PC WAS PUSHED ON STACK
3634  011766  000200                      RTS     R0              ;GO BACK TO INDICATE BAD ABORT
3635
```

```
3636                                    ;*******************************************************************************
3637                                    ;END OF PASS ROUTINE
3638                                    ;INCREMENT THE PASS NUMBER
3639                                    ;IF SW10=0 DING THE TTY BELL ON END OF PROGRAM
3640                                    ;IF THERE IS A MONITOR GO TO IT.
3641                                    ;IF NONE JUMP TO START
3642  011770  000004                    $EOP:   SCOPE
3643  011772  005067  167304                   CLR     $TSTNM                  ;ZERO THE TEST NUMBER
3644  011776  005067  000306                   CLR     $TIMES                  ;ZERO THE NUMBER OF ITERATIONS
3645  012002  005267  167272                   INC     $PASS                   ;INCREMENT THE PASS NUMBER
3646  012006  032737  002000  177570          BIT     #SW10,@#SWR             ;RING THE BELL?
3647  012014  001002                          BNE     4$                      ;NO
3648  012016  104400  012056                 TYPE    $BELL                   ;RING A DING
3649  012022  013700  000042          4$:     MOV     @#42,R0                 ;GET MONITOR ADDRESS
3650  012026  001411                          BEQ     $DOAGN                  ;IF NONE
3651  012030  022760  177777  000002          CMP     #-1,2(R0)
3652  012036  001001                          BNE     $ENDAD
3653  012040  000005                          RESET
3654  012042  004710                  $ENDAD: JSR     PC,(R0)                 ;GO TO MONITOR
3655  012044  000240                          NOP                             ;SAVE ROOM
3656  012046  000240                          NOP                             ;FOR
3657  012050  000240                          NOP                             ;ACT11
3658  012052  000137  003620          $DOAGN: JMP     @#START                 ;RETURN
3659  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
012062                          $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     @#ERRVEC,-(SP)          ;SAVE THE CONTENTS OF THE ERROR VECTOR
012076  012737  012116  000004      MOV     #5$,@#ERRVEC            ;SET FOR TIMEOUT
012104  005737  177060              TST     @#177060                ;TIME OUT ON XOR?
012110  012637  000004              MOV     (SP)+,@#ERRVEC          ;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)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
012124  000417                      BR      7$                      ;LOOP ON THE PRESENT TEST
012126                          6$: ;#####END OF CODE FOR THE XOR TESTER#####
012126  032737  000400  177570      BIT     #SW08,@#SWR             ;LOOP ON SPEC. TEST?
012134  001404                      BEQ     2$                      ;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
```

```
                                ;*************************************************************************
                                ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
                                ;CALL:
                                ;       ACCEPT  ,ADDR           ;PUT OCTAL NUMBER IN ADDR
                                ;
012314                          $ACCEPT:
012314  010046                          MOV     R0,-(SP)                ;PUSH R0 ON STACK
012316  010146                          MOV     R1,-(SP)                ;PUSH R1 ON STACK
012320  010246                          MOV     R2,-(SP)                ;PUSH R2 ON STACK
012322  010346                          MOV     R3,-(SP)                ;PUSH R3 ON STACK
012324  016600  000010                  MOV     10(SP),R0               ;GET ADDRESS POINTER OF WHERE TO PUT NUMBER
012330  005001                  1$:     CLR     R1              ;CLEAR PARTIAL NUMBER
012332  012702  000006                  MOV     #6,R2           ;MAX. # OF DIGITS ALLOWED
012336  104402                  2$:     GETCHR                  ;GET ONE CHARACTER
012340  112603                          MOVB    (SP)+,R3        ;AND PUT IT IN R3
012342  110367  000102                  MOVB    R3,6$
012346  104400  012450                  TYPE    ,6$             ;ECHO THE CHARACTER
012352  022703  000015                  CMP     #15,R3          ;WAS THIS CHARACTER A "CR"?
012356  001422                          BEQ     5$              ;BR IF YES
012360  022703  000060                  CMP     #'0,R3          ;INSURE THE CHARACTER IS
012364  003014                          BGT     4$              ;A DIGIT BETWEEN 0 AND 7.
012366  022703  000067                  CMP     #'7,R3
012372  002411                          BLT     4$
012374  005302                          DEC     R2              ;CHECK NUMBER OF CHARACTERS
012376  002407                          BLT     4$              ;BR IF TO MANY
012400  006301                          ASL     R1              ;POSITION PARTIAL NUMBER
012402  006301                          ASL     R1              ;FOR THIS DIGIT
012404  006301                          ASL     R1
012406  042703  177770                  BIC     #^C<7>,R3       ;GET RID OF THE ASCII JUNK
012412  050301                          BIS     R3,R1           ;COMBINE THIS DIGIT WITH PARTIAL
012414  000750                          BR      2$              ;GO GET ANOTHER DIGIT
012416  104400  012452          4$:     TYPE    ,$QUES          ;TYPE "?"
012422  000742                          BR      1$              ;GO START OVER
012424  104400  012454          5$:     TYPE    ,$LF            ;FOLLOW "CR" WITH A "LF"
012430  010130                          MOV     R1,@(R0)+       ;PASS THE NUMBER TO THE USER
012432  010066  000010                  MOV     R0,10(SP)       ;SET FOR RETURN
012436  012603                          MOV     (SP)+,R3                ;POP STACK INTO R3
012440  012602                          MOV     (SP)+,R2                ;POP STACK INTO R2
012442  012601                          MOV     (SP)+,R1                ;POP STACK INTO R1
012444  012600                          MOV     (SP)+,R0                ;POP STACK INTO R0
012446  000002                          RTI
012450     000      000         6$:     .BYTE   0,0             ;STORAGE FOR ASCII CHAR. AND TERMINATOR
012452     077                  $QUES:  .ASCII  "?"                     ;QUESTION MARK
012453     015                  $CRLF:  .ASCII  <15>                    ;CARRAIGE RETURN
012454  000012                  $LF:    .ASCIZ  <12>                    ;LINEFEED
                                                                        ;SELECTED REGISTER FROM THE TTY
                                ;*************************************************************************
                                ;THIS ROUTINE INPUTS A SINGLE CHARACTER FROM THE TTY
                                ;CALL:
                                ;
                                ;       GETCHR                  ;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;       RETURN HERE             ;CHARACTER IS ON THE STACK
                                ;
012456  011646                  $READC: MOV     (SP),-(SP)      ;PUSH DOWN THE PC
012460  016666  000004  000002          MOV     4(SP),2(SP)     ;SAVE THE PS
012466  105777  000126          1$:     TSTB    @TKS            ;WAIT FOR
```

```
3770  012472  100375                        BPL     1$                      ;A CHARACTER
3771  012474  117766  000122  000004        MOVB    @TKB,4(SP)              ;READ THE TTY
3772  012502  042766  177600  000004        BIC     #^C<177>,4(SP)          ;GET RID OF JUNK IF ANY
3773  012510  000002                        RTI                             ;GO BACK TO USER
3774                                ;******************************************************************************
3775                                ;THIS ROUTINE INPUTS A STRING FROM THE TTY
3776                                ;CALL:
3777                                ;       GETSTR                          ;INPUT A STRING FROM THE TTY
3778                                ;       RETURN HERE                     ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3779                                ;                                       ;TERMINATOR WILL BE A BINARY 0
3780                                ;
3781  012512  010346                $READS: MOV     R3,-(SP)                ;SAVE R3
3782  012514  012703  012624        1$:     MOV     #$TTYIN,R3              ;GET ADDRESS
3783  012520  022703  012634        2$:     CMP     #$TTYIN+8.,R3           ;BUFFER FULL?
3784  012524  101405                        BLOS    4$                      ;BR IF YES
3785  012526  104402                        GETCHR                          ;GO READ ONE CHARACTER FROM THE TTY
3786  012530  112613                        MOVB    (SP)+,(R3)              ;GET CHARACTER
3787  012532  122713  000177                CMPB    #177,(R3)               ;IS IT A RUBOUT
3788  012536  001003                        BNE     3$                      ;SKIP IF NOT
3789  012540  104400  012452        4$:     TYPE    $QUES                   ;TYPE A '?'
3790  012544  000763                        BR      1$                      ;ZAP THE BUFFER AND LOOP
3791  012546  111367  000044        3$:     MOVB    (R3),8$                 ;ECHO THE CHARACTER
3792  012552  104400  012616                TYPE    ,8$
3793  012556  122723  000015                CMPB    #15,(R3)+               ;CHECK FOR RETURN
3794  012562  001356                        BNE     2$                      ;LOOP IF NOT RETURN
3795  012564  105063  177777                CLRB    -1(R3)                  ;ZAP RETURN (THE 15)
3796  012570  104400  012454                TYPE    $LF                     ;TYPE A LINE FEED
3797  012574  012603                        MOV     (SP)+,R3                ;RESTORE R3
3798  012576  011646                        MOV     (SP),-(SP)              ;ADJUST THE STACK AND PUT ADDRESS OF THE
3799  012600  016666  000004  000002        MOV     4(SP),2(SP)             ;       FIRST ASCII CHARACTER ON IT
3800  012606  012766  012624  000004        MOV     #$TTYIN,4(SP)
3801  012614  000002                        RTI
3802  012616     000                8$:     .BYTE   0                       ;RETURN
3803  012617     000                        .BYTE   0                       ;STORAGE FOR ASCII CHAR.. TO TYPE
3804  012620  177560                TKS:    177560                          ;TERMINATOR
3805  012622  177562                TKB:    177562                          ;TTY KBD STATUS
3806  012624  000010                $TTYIN: .BLKB   8.                      ;TTY KBD BUFFER
3807                                                                        ;RESERVE 8 BYTES FOR TTY INPUT
                                                                                ;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'
```

```
3864  013034  000000                    1$:   0                       ;'ERROR MESSAGE' POINTER GOES HERE
3865  013036  104400  012453                  TYPE  ,$CRLF            ;TYPE A CARRIAGE RETURN AND
3866                                                                    ;LINE FEED
3867  013042  012067  000004                  MOV   (R0)+,2$          ;PICK UP 'DATA HEADER' POINTER
3868  013046  001402                          BEQ   3$                ;IF '0' DON'T TYPE
3869  013050  104400                          TYPE                    ;TYPE 'DATA HEADER'
3870  013052  000000                    2$:   0                       ;'DATA HEADER' POINTER GOES HERE
3871  013054  104400  012453              3$:   TYPE  ,$CRLF            ;TYPE A CARRIAGE RETURN AND
3872                                                                    ;LINE FEED
3873  013060  012000                          MOV   (R0)+,R0          ;PICK UP 'DATA POINTER'
3874  013062  001004                          BNE   5$                ;IF THERE IS DATA TO TYPE GO DO IT
3875  013064  012600                    4$:   MOV   (SP)+,R0          ;RESTORE R0
3876  013066  104400  012453                  TYPE  ,$CRLF            ;TYPE A CARRIAGE RETURN AND
3877                                                                    ;AND LINE FEED
3878  013072  000207                          RTS   PC                ;RETURN TO TESTING
3879  013074                             5$:
3880  013074  013046                          MOV   @(R0)+,-(SP)      ;SAVE @(R0)+ FOR TYPEOUT
3881                                                                    ;TYPE DATA
3882  013076  004067  000246                  JSR   R0,$B2OCT         ;GO TYPE--OCTAL ASCII
3883  013102     006                          .BYTE 6                 ;TYPE 6 DIGITS
3884  013103     001                          .BYTE 1                 ;TYPE LEADING ZEROS
3885  013104  005710                          TST   (R0)              ;HAVE WE REACHED THE '0' TERMINATOR
3886  013106  001766                          BEQ   4$                ;YES - CLEAN UP FOR RETURN
3887  013110  104400  013116                  TYPE  ,6$               ;TYPE 5 SPACES
3888  013114  000767                          BR    5$                ;LOOP TILL '0' TERMINATOR REACHED
3889  013116  020040  020040  000040  6$:   .ASCIZ /     /
3890                                          .EVEN
```

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

```
3947                                    ;       JSR     R0,$B20CT               ;CALL FOR TYPEOUT
3948                                    ;       .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3949                                    ;       .BYTE   M                       ;M=1 OR 0
3950                                    ;                                               ;1=TYPE LEADING ZEROS
3951                                    ;                                               ;0=SUPPRESS LEADING ZEROS
3952                                    ;
3953                                    ;$B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B20CT OR $B2016
3954                                    ;CALL:
3955                                    ;       MOV     NUM,-(SP)
3956                                    ;       JSR     R0,$B201
3957                                    ;
3958                                    ;$B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3959                                    ;CALL:
3960                                    ;       MOV     NUM,-(SP)
3961                                    ;       JSR     R0,$B2016
3962
3963  013350  112067  000201          $B20CT: MOVB    (R0)+,$OMODE+1          ;PICKUP THE NUMBER OF DIGITS TO TYPE
3964  013354  112067  000173                  MOVB    (R0)+,$OFILL            ;GET THE ZERO FILL SWITCH
3965  013360  000406                          BR      $B201
3966  013362  112767  000001  000163 $B2016: MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
3967  013370  112767  000006  000157         MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
3968  013376  112767  000005  000146 $B201:  MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
3969  013404  010346                          MOV     R3,-(SP)                ;SAVE R3
3970  013406  010446                          MOV     R4,-(SP)                ;SAVE R4
3971  013410  010546                          MOV     R5,-(SP)                ;SAVE R5
3972  013412  116704  000137                  MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
3973  013416  005404                          NEG     R4
3974  013420  062704  000006                  ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
3975  013424  110467  000124                  MOVB    R4,$OMODE               ;SAVE IT FOR USE
3976  013430  116704  000117                  MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
3977  013434  016605  000010                  MOV     10(SP),R5               ;PICKUP THE INPUT NUMBER
3978  013440  005003                          CLR     R3                      ;CLEAR THE OUTPUT WORD
3979  013442  006105                  1$:     ROL     R5                      ;ROTATE MSB INTO "C"
3980  013444  000404                          BR      3$                      ;GO DO MSB
3981  013446  006105                  2$:     ROL     R5                      ;FORM THIS DIGIT
3982  013450  006105                          ROL     R5
3983  013452  006105                          ROL     R5
3984  013454  010503                          MOV     R5,R3
3985  013456  006103                  3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
3986  013460  105367  000070                  DECB    $OMODE                  ;TYPE THIS DIGIT?
3987  013464  100016                          BPL     7$                      ;BR IF NO
3988  013466  042703  177770                  BIC     #177770,R3              ;GET RID OF JUNK
3989  013472  001002                          BNE     4$                      ;TEST FOR 0
3990  013474  005704                          TST     R4                      ;SUPPRESS THIS 0?
3991  013476  001403                          BEQ     5$                      ;BR IF YES
3992  013500  005204                  4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
3993  013502  052703  000060                  BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
3994  013506  052703  000040          5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
3995  013512  110367  000032                  MOVB    R3,8$                   ;SAVE FOR TYPING
3996  013516  104400  013550                  TYPE    ,8$                     ;GO TYPE THIS DIGIT
3997  013522  105367  000024          7$:     DECB    $OCNT                   ;COUNT BY 1
3998  013526  003347                          BGT     2$                      ;BR IF MORE TO DO
3999  013530  002402                          BLT     6$                      ;BR IF DONE
4000  013532  005204                          INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
4001  013534  000744                          BR      2$                      ;GO DO THE LAST DIGIT
4002  013536  012605                  6$:     MOV     (SP)+,R5                ;RESTORE R5
```

```
 4003  013540  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4004  013542  012603                        MOV     (SP)+,R3        ;RESTORE R3
 4005  013544  012616                        MOV     (SP)+,(SP)      ;SET THE STACK FOR RETURNING
 4006  013546  000200                        RTS     R0              ;RETURN
 4007  013550     000              8$:       .BYTE   0               ;STORAGE FOR ASCII DIGIT
 4008  013551     000                        .BYTE   0               ;TERMINATOR FOR TYFE ROUTINE
 4009  013552     000              $OCNT:    .BYTE   0               ;OCTAL DIGIT COUNTER
 4010  013553     000              $OFILL:   .BYTE   0               ;ZERO FILL SWITCH
 4011  013554  000000              $OMODE:   0                       ;NUMBER OF DIGITS TO TYPE
```

```
4012
4013                                    ;*****************************************************************************
4014                                    ;TRAP HANDLER
4015  013556  010046                    $TRAP:  MOV     R0,-(SP)        ;SAVE R0
4016  013560  016600  000002                    MOV     2(SP),R0        ;GET TRAP ADDRESS
4017  013564  005740                            TST     -(R0)           ;BACKUP BY 2
4018  013566  111000                            MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP
4019  013570  016000  013576                    MOV     $TRPAD(R0),R0   ;INDEX TO TABLE
4020  013574  000200                            RTS     R0              ;GO TO ROUTINE
4021
4022
4023                                    ;TRAP TABLE
4024                                    ;       ROUTINE
4025                                    ;       -------
4026  013576  001110                    $TRPAD: $TYPE   ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
4027  013600  012456                            $READC  ;CALL=GETCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
4028  013602  012512                            $READS  ;CALL=GETSTR    TRAP+4(104404)  TTY TYPEIN STRING ROUTINE
4029  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        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
```

```
4068                                    ;*****************************************************************************
4069                                    ;
4070                                    ;ERROR AND MESSAGE TABLE CONDIMENTS
4071                                    ;
4072                                    ;*****************************************************************************
4073
4074  013752  042524  052123  042040  EM1:    .ASCIZ  /TEST DIDN'T ABORT  /
4075  013760  042111  023516  020124
4076  013766  041101  051117  020124
4077  013774  000040
4078  013776  040506  040524  020114  EM2:    .ASCIZ  /FATAL ERROR TO PROGRAM  /
4079  014004  051105  047522  020122
4080  014012  047524  050040  047522
4081  014020  051107  046501  020040
4082  014026     000
4083  014027     101  047502  052122  EM3:    .ASCIZ  /ABORTED INCORRECTLY  /
4084  014034  042105  044440  041516
4085  014042  051117  042522  052103
4086  014050  054514  020040     000
4087  014055     116  020117  040520  EM4:    .ASCIZ  /NO PARITY MEMORY FOUND BELOW 28K  /
4088  014062  044522  054524  046440
4089  014070  046505  051117  020131
4090  014076  047506  047125  020104
4091  014104  042502  047514  020127
4092  014112  034062  020113  000040
4093  014120  042522  042523  020124  EM5:    .ASCIZ  /RESET DOESN'T WORK  /
4094  014126  047504  051505  023516
4095  014134  020124  047527  045522
4096  014142  020040     000
4097  014145     125  042523  020122  EM6:    .ASCIZ  /USER SELECTED REGISTER NOT PRESENT  /
4098  014152  042523  042514  052103
4099  014160  042105  051040  043505
4100  014166  051511  042524  020122
4101  014174  047516  020124  051120
4102  014202  051505  047105  020124
4103  014210  000040
4104  014212  047516  050040  051101  EM7:    .ASCIZ  /NO PARITY MEMORY FOUND AT ALL../
4105  014220  052111  020131  042515
4106  014226  047515  054522  043040
4107  014234  052517  042116  040440
4108  014242  020124  046101  020114
4109  014250  000040
4110  014252  044504  047104  052047  EM10:   .ASCIZ  /DIDN'T ABORT OR RECOGNIZE STACK VIOLATION  /
4111  014260  040440  047502  052122
4112  014266  047440  020122  042522
4113  014274  047503  047107  055111
4114  014302  020105  052123  041501
4115  014310  020113  044526  046117
4116  014316  052101  047511  020116
4117  014324  000040
4118  014326  041101  051117  042524  EM11:   .ASCIZ  /ABORTED BUT STACK VIOLATION NOT RECOGNIZED  /
4119  014334  020104  052502  020124
4120  014342  052123  041501  020113
4121  014350  044526  046117  052101
4122  014356  047511  020116  047516
4123  014364  020124  042522  047503
```

```
 4124  014372  047107  055111  042105
 4125  014400  020040     000
 4126  014403     123  040524  045503  EM12:   .ASCIZ  /STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED  /
 4127  014410  053040  047511  040514
 4128  014416  044524  047117  050040
 4129  014424  041511  042513  020104
 4130  014432  050125  041040  052125
 4131  014440  040440  047502  052122
 4132  014446  047040  052117  051040
 4133  014454  041505  043517  044516
 4134  014462  042532  020104  000040
 4135  014470  051120  043517  040522  DH1:    .ASCII  /PROGRAM   REGISTER/<15><12>
 4136  014476  020115  020040  042522
 4137  014504  044507  052123  051105
 4138  014512  005015
 4139  014514  020040  041520  020040          .ASCIZ  /  PC      UNDER TEST/
 4140  014522  020040  052440  042116
 4141  014530  051105  052040  051505
 4142  014536  000124
 4143  014540  051040  043505  051511  DH2:    .ASCII  / REGISTER/<15><12>
 4144  014546  042524  006522     012
 4145  014553     125  042116  051105          .ASCIZ  /UNDER TEST/
 4146  014560  052040  051505  000124
 4147  014566  051120  043517  040522  DH3:    .ASCII  /PROGRAM/<15><12>
 4148  014574  006515     012
 4149  014577     040  050040  000103          .ASCIZ  /  PC/
 4150  014604  051120  043517  040522  DH4:    .ASCII  /PROGRAM   REGISTER    EXPECTED     ACTUAL    EXPECTED   ACTUAL/<15><12>
 4151  014612  020115  020040  042522
 4152  014620  044507  052123  051105
 4153  014626  020040  020040  054105
 4154  014634  042520  052103  042105
 4155  014642  020040  020040  040440
 4156  014650  052103  040525  020114
 4157  014656  020040  042440  050130
 4158  014664  041505  042524  020104
 4159  014672  020040  041501  052524
 4160  014700  046101  005015
 4161  014704  020040  041520  020040          .ASCIZ  /  PC     UNDER TEST  ADDR. BITS  ADDR. BITS  ABORT PC   ABORT PC/
 4162  014712  020040  052440  042116
 4163  014720  051105  052040  051505
 4164  014726  020124  040440  042104
 4165  014734  027122  041040  052111
 4166  014742  020123  040440  042104
 4167  014750  027122  041040  052111
 4168  014756  020123  040440  047502
 4169  014764  052122  050040  020103
 4170  014772  020040  041101  051117
 4171  015000  020124  041520     000
 4172  015005     120  047522  051107  DH5:    .ASCII  /PROGRAM   REGISTER    EXPECTED/<15><12>
 4173  015012  046501  020040  051040
 4174  015020  043505  051511  042524
 4175  015026  020122  020040  042440
 4176  015034  050130  041505  042524
 4177  015042  006504     012
 4178  015045     040  050040  020103          .ASCIZ  /  PC      UNDER TEST   ABORT PC/
 4179  015052  020040  020040  052440
```

```
 4180  015060  042116  051105  052040
 4181  015066  051505  020124  020040
 4182  015074  041101  051117  020124
 4183  015102  041520     000
 4184          015106                        .EVEN
 4185  015106  001324  001622  000000  DT1:  .WORD   $HLTAD,PARITY,0
 4186  015114  001622  000000          DT2:  .WORD   PARITY,0
 4187  015120  001324  000000          DT3:  .WORD   $HLTAD,0
 4188  015124  001324  001622  001326  DT4:  .WORD   $HLTAD,PARITY,$GDADR,$BDADR,$GDDAT,$BDDAT,0
 4189  015132  001330  001332  001334
 4190  015140  000000
 4191  015142  001324  001622  001332  DT5:  .WORD   $HLTAD,PARITY,$GDDAT,0
 4192  015150  000000
 4193
 4194          000001                .END
```

DCKBRE.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
A       003752          1710    1720#
ABORT   003430          1368    1567#
ACCEPT= 104406          1322    4029#
AD      004026          1732    1742#
A1      004104          1754    1764#
A2      004162          1776    1786#
B       004236          1798    1808#
BEGIN   001706           851    1207#  4060
BITO  = 000001           828#   1503   1504   1511   1512   1587   1593   1720   1742   1764   1786   1808   1830
                        1852    1874   1899   1921   1946   1969   1991   2013   2038   2063   2085   2107   2129
                        2151    2173   2198   2220   2242   2264   2375   2474   2526   2573   2620   2677   2733
                        2736    2738   2763   2785   2807   2829   2851   2942   3006   3086   3189   3239   3270
                        3301    3351   3401   3451   3459   3484   3817
BIT00 = 000001           818#    828   1208   1209   3643   3660   3759   3807   3843
BIT01 = 000002           817#    827   1208   1212
BIT02 = 000004           816#    826   1208   1214
BIT03 = 000010           815#    825   1208   1216
BIT04 = 000020           814#    824
BIT05 = 000040           813#    823
BIT06 = 000100           812#    822
BIT07 = 000200           811#    821
BIT08 = 000400           810#    820
BIT09 = 001000           809#    819
BIT1  = 000002           827#
BIT10 = 002000           808#
BIT11 = 004000           807#
BIT12 = 010000           806#   1314
BIT13 = 020000           805#
BIT14 = 040000           804#    856
BIT15 = 100000           803#   3459
BIT2  = 000004           826#   1519   1520   1523   1524   1587   1593   1717   1720   1739   1742   1761   1764
                        1783    1786   1805   1808   1827   1830   1849   1852   1871   1874   1889   1891   1896
                        1899    1918   1921   1935   1937   1943   1946   1966   1969   1988   1991   2010   2013
                        2028    2030   2035   2038   2052   2054   2060   2063   2082   2085   2104   2107   2126
                        2129    2148   2151   2170   2173   2187   2189   2195   2198   2217   2220   2239   2242
                        2261    2264   2317   2346   2355   2435   2451   2453   2490   2508   2514   2542   2551
                        2555    2609   2614   2618   2675   2733   2760   2763   2782   2785   2804   2807   2826
                        2829    2848   2851   2929   2935   2937   3001   3060   3163   3221   3226   3229   3252
                        3257    3260   3283   3288   3291   3314   3326   3344   3364   3376   3394   3414   3426
                        3444    3459   3484
BIT3  = 000010           825#
BIT4  = 000020           824#
BIT5  = 000040           823#
BIT6  = 000100           822#
BIT7  = 000200           821#
BIT8  = 000400           820#
BIT9  = 001000           819#
BLKCNT  001632          1172#   1353*  1373*  1375   1389*  1397*  1405*  1407   1421*  1429*
BPTVEC= 000014           835#
BO      004312          1820    1830#
B1      004370          1842    1852#
B2      004444          1864    1874#
B3      004536          1886    1899#
B4      004612          1911    1921#
B5      004706          1933    1946#
B6      004766          1958    1969#
```

```
C       005042          1981    1991#
CHECKL  011550          1721    1743    1765    1787    1909    1831    1853    1875    1900    1922    1947    1970    1992
                        2014    2039    2064    2086    2108    2130    2152    2174    2199    2221    2243    2265    2376
                        2475    2527    2574    2621    2678    2739    2764    2786    2808    2830    2852    2943    3007
                        3098    3201    3240    3271    3302    3352    3402    3452    3546#
COMPUT  003204          1482#   1571    1693
CPU40   001642          1190#   1253*   1262*   2360    2436    2496    2560    2990    3327    3377    3427
C0      005116          2003    2013#
C1      005210          2025    2038#
C2      005304          2050    2063#
C3      005360          2075    2085#
C4      005436          2097    2107#
C5      005514          2119    2129#
C6      005570          2141    2151#
C7      005644          2163    2173#
C8      005742          2185    2198#
D       006020          2210    2220#
DD      006006          2212    2217#
DD0     006064          2234    2239#
DD1     006144          2256    2261#
DH1     014470          1043    1064    1089    1096    4135#
DH2     014540          1059    1075    4143#
DH3     014566          1070    4147#
DH4     014604          1049    4150#
DH5     015[illegible]05          1036    1081    4172#
DISPLA= 177[illegible]70          760#    3709*
DT1     015106          1045    1066    1091    1098    4185#
DT2     015114          1060    1076    4186#
DT3     015120          1071    4187#
DT4     015124          1055    4188#
DT5     015142          1039    1084    4191#
D0      006076          2232    2242#
D1      006156          2254    2264#
E       006320          2372#
EMTVEC= 000030           838#   1212*   1213*
EM1     013752          1035    4074#
EM10    014252          1079    4110#
EM11    014326          1087    4118#
EM12    014403          1094    4126#
EM2     013776          1042    4078#
EM3     014027          1048    4083#
EM4     014055          1058    4087#
EM5     014120          1063    4093#
EM6     014145          1069    4097#
EM7     014212          1074    4104#
ERRVEC= 000004           831#   1266*   1267*   3066*   3069*   3169*   3172*   3674    3675*   3677*   3680*   3929*   3935*
E0      006326          2316    2375#
E1      006502          2448    2471#
E2      006510          2430    2474#
F       006634          2520#   2525
FINDON  002504          1316    1328#
FLAGSC  011474          3472    3499#
F0      006644          2489    2526#
F1      006774          2538    2573#
G       007056          2610    2618#
GETCHR= 104402          3727    3785    4027#
```

```
GETSTR= 104404          4028#
GO      002326          1276    1283    1287#
GO      007066          2608    2620#
H       007140          2672    2675#
HOLDLO  010302          3167#   3204
HO      007150          2666    2677#
H1      007134          2668    2669    2673#
H2      007242          2724    2738#
H3      007216          2726    2727    2731#
H4      007222          2730    2733#
INITIA  011506          1289    3491    3510#
INTERT  001644          1196#   1331    1395    1427
INTVEC  001620          1160#   3483*
IOTVEC= 000020           836#   1209*   1210*
KPAR0   000230           878#   1281*   3912*
KPAR1   000232           879#   1335*   1362*   1687*   3913*
KPAR2   000234           880#   3914*
KPAR3   000236           881#   3915*
KPAR4   000240           882#   3916*
KPAR5   000242           883#   3917*
KPAR6   000244           884#   3918*   3922
KPAR7   000246           885#   3919*   3931
KPDR0   000210           867#   3904*
KPDR1   000212           868#   3905*
KPDR2   000214           869#   3906*
KPDR3   000216           870#   3907*
KPDR4   000220           871#   3908*
KPDR5   000222           872#   3909*
KPDR6   000224           873#   3910*
KPDR7   000226           874#   3911*
KTTIME  002324          1266    1284#
L       007320          2753    2763#
LEAFCN  001636          1185#   1251*   1391*   1399*   1423*   1431*   1596*   1597    1622*
M       007374          2775    2785#
MEMAD   001640          1188#   1352*   1358*   1360    1364*   1365    1366    1367    1378*   1381*   1403    1410*   1413*
MRK0    006324          2371    2374#
MRK1    006506          2470    2473#
MSGTYP  002372          1317#   1441    1454    1464
MSREGF  001626          1166#   1554*   1623*   3501*   3512*   3553
N     = 000067           840#   1499    1503#   1507    1511#   1515    1519#   1527    1531#   1542    1546#   1702    1706#
                        1724    1728#   1746    1750#   1768    1772#   1790    1794#   1812    1816#   1834    1838#   1856
                        1860#   1878    1882#   1903    1907#   1925    1929#   1950    1954#   1973    1977#   1995    1999#
                        2017    2021#   2042    2046#   2067    2071#   2089    2093#   2111    2115#   2133    2137#   2155
                        2159#   2177    2181#   2202    2206#   2224    2228#   2246    2250#   2308    2312#   2422    2426#
                        2481    2485#   2530    2534#   2600    2604#   2658    2662#   2716    2720#   2745    2749#   2767
                        2771#   2789    2793#   2811    2815#   2833    2837#   2855    2859#   2920    2924#   2978    2982#
                        3048    3052#   3152    3156#   3210    3214#   3243    3247#   3274    3278#   3305    3309#   3355
                        3359#   3405    3409#   3455    3459#
NEWSTK  001476          1016#   1695*   2333    2378    2440    2477    2495    2529    2576    2623    2680
NEXT1   002514          1327    1331#   1445    1458    1467    1624
NN      007450          2797    2807#
NOMORE  003172          1341    1469#
NOREG   003144          1338    1459#
NTERAD  001446          1004#   1295    2335    2347    2431    2460    2540    2879    3470*   3526*
NTERO   001450          1005#   1240*   3526
NTER1   001452          1006#   1241*
```

```
NTER10  001470          1013#  1248*
NTER11  001472          1014#  1249*
NTER12  001474          1015#  1250*
NTER2   001454          1007#  1242*
NTER3   001456          1008#  1243*
NTER4   001460          1009#  1244*
NTER5   001462          1010#  1245*
NTER6   001464          1011#  1246*
NTER7   001466          1012#  1247*
ONETRY  003500          1583   1593#
P       007526          2819   2829#
PARCOR  003112          1350   1446#
PARITY  001622          1161#  1435*  1447*  1498*  1503*  1504   1511*  1512   1519*  1520   1523*  1524   1535*
                        1537   1539*  1547*  1550   1573*  1587*  1593*  1679*  1717*  1720*  1739*  1742*  1761*
                        1764*  1783*  1786*  1805*  1808*  1827*  1830*  1849*  1852*  1871*  1874*  1889*  1891*
                        1896*  1899*  1918*  1921*  1935*  1937*  1943*  1946*  1966*  1969*  1988*  1991*  2010*
                        2013*  2028*  2030*  2035*  2038*  2052*  2054*  2060*  2063*  2082*  2085*  2104*  2107*
                        2126*  2129*  2148*  2151*  2170*  2173*  2187*  2189*  2195*  2198*  2217*  2220*  2239*
                        2242*  2261*  2264*  2317*  2346*  2355*  2375*  2435*  2451*  2453*  2474*  2490*  2508*
                        2514*  2526*  2542*  2551*  2555*  2573*  2609*  2614*  2618*  2620*  2675*  2677*  2733*
                        2736*  2738*  2760*  2763*  2782*  2785*  2804*  2807*  2826*  2829*  2848*  2851*  2929*
                        2935*  2937*  2942*  3001*  3006*  3060*  3086*  3163*  3189*  3221*  3226*  3229*  3239*
                        3252*  3257*  3260*  3270*  3283*  3288*  3291*  3301*  3314*  3326*  3344*  3351*  3364*
                        3376*  3394*  3401*  3414*  3426*  3444*  3451*  3459*  3484*  3581   3817*  4185   4186
                        4188   4191
PARTST  003222          1343   1498#
PC     =%000007          772#   930*   937*   942*  1277*  1280*  1699*  2371*  2470*  3062*  3165*  3654*  3829*
                        3878*  3925   3940*
PS     = 177776          757#   758
PSPCOR  001624          1162#  2886*  3217*  3499*  3510*  3557
PSW    = 177776          758#
PWRVEC= 000024           837#  1216*  1217*  4033*  4034*  4042*  4057*  4058*
R       007604          2841   2851#
RED     010400          2885   3183   3187   3195   3203#
RESTAR  011464          1678   3491#
RESTOR  001634          1177#  1360*  1365*  1381   1413
RESVEC= 000010           832#  1256*  1257*  2930*  2936*  2987*  2989*
R0     =%000000          763#   922    923*   924    927*  1271*  1272*  1273   1292*  1361*  1362   1482*  1484
                        1486   1487   1488*  1568   1569*  1582*  1584*  1585*  1586*  1588*  1594*  1602*  1614
                        1691*  1709*  1711*  1712*  1716*  1721*  1731*  1733*  1734*  1738*  1743*  1753*  1755*
                        1756*  1760*  1765*  1775*  1777*  1778*  1782*  1787*  1797*  1799*  1800*  1804   1809*
                        1819*  1821*  1822*  1826   1831*  1841*  1843*  1844*  1848   1853*  1863*  1865*  1866*
                        1870   1875*  1885*  1887*  1888*  1890*  1895   1900*  1910*  1912*  1913*  1917   1922*
                        1932*  1934*  1936*  1938*  1942   1947*  1957*  1959*  1960*  1961*  1965   1970*  1980*
                        1982*  1983*  1987   1992*  2002*  2004*  2005*  2009   2014*  2024*  2026*  2027*  2029*
                        2034   2039*  2049*  2051*  2053*  2055*  2059   2064*  2074*  2076*  2077*  2081   2086*
                        2096*  2098*  2099*  2103   2108*  2118*  2120*  2121*  2125   2130*  2140*  2142*  2143*
                        2147   2152*  2162*  2164*  2165*  2169   2174*  2184*  2186*  2188*  2190*  2194   2199*
                        2209*  2211*  2212*  2216   2221*  2231*  2233*  2234*  2238   2243*  2253*  2255*  2256*
                        2260   2265*  2315*  2376*  2429*  2440*  2441*  2444   2475*  2488*  2491*  2525*  2527*
                        2537*  2539*  2548*  2552*  2556*  2564*  2567   2574*  2607*  2610*  2615   2621*  2665*
                        2678*  2723*  2739*  2752*  2754*  2755*  2759*  2764*  2774*  2776*  2777*  2781   2786*
                        2796*  2798*  2799*  2803*  2808*  2818*  2820*  2821*  2825   2830*  2840*  2842*  2843*
                        2847*  2852*  2927*  2943*  2985*  3007*  3055*  3098*  3159*  3201*  3219*  3222*  3223*
                        3227*  3230   3233*  3236   3240*  3250*  3253*  3254*  3258*  3261   3264*  3267   3271*
                        3281*  3284*  3285*  3289*  3292   3295*  3298   3302*  3312*  3352*  3362*  3402*  3412*
```

```
                    3452*  3483   3485*  3606*  3607   3609*  3631*  3634*  3649*  3651   3654   3720   3724*
                    3748*  3749   3753*  3854   3855*  3856*  3857*  3858*  3859*  3860*  3861*  3862   3867
                    3873*  3875*  3880   3882*  3885   3933*  3934*  3937   3963   3964   4006*  4015   4016*
                    4017   4018*  4019*  4020*  4035   4056*
R1      =%000001     764#  1484*  1487*  1488   1584   1695   1711   1733   1755   1777   1799   1821   1843
                    1865   1887   1890   1912   1934   1938   1959   1982   2004   2026   2029   2051   2055
                    2076   2098   2120   2142   2164   2186   2190   2211   2233   2255   2494   2539   2570
                    2754   2776   2798   2820   2842   2859   3222   3253   3284   3315   3365   3415   3578
                    3721   3725*  3739*  3740*  3741*  3743*  3748   3752*  4036   4055*
R2      =%000002     765#  1291*  1328*  1339   1342   1435   1443*  1447   1456*  1465*  1470*  1498   1573
                    1618*  1619*  2335*  2337*  2340   2343*  2347*  2349*  2352*  2354   2358*  2365*  2368
                    2431*  2433*  2454   2457*  2460*  2462*  2465*  2467   2494*  2509*  2515*  2520   2540*
                    2543   2546*  3315*  3316*  3318*  3321*  3323*  3331   3336*  3339*  3341   3345*  3348
                    3365*  3366*  3368*  3371*  3373*  3381   3386*  3389*  3391   3395*  3398   3415*  3416*
                    3418*  3421*  3423*  3431   3436*  3439*  3441   3445*  3448   3546   3549*  3560*  3568*
                    3575*  3587   3590   3596*  3612*  3616   3622*  3722   3726*  3737*  3751*  4037   4054*
R3      =%000003     766#  1289*  1290*  1367*  1489*  1569   1571*  1612   1693*  2495*  2498*  2502*  2505
                    2520*  2570*  2615*  2619*  3236*  3267*  3298*  3348*  3398*  3448*  3472*  3491*  3503*
                    3530*  3547   3573*  3578   3595*  3610*  3621*  3723   3728*  3729   3731   3733   3735
                    3742*  3743   3750*  3781   3782*  3783   3786*  3787   3791   3793   3795*  3797*  3903*
                    3904   3905   3906   3907   3908   3909   3910   3911   3969   3978*  3984*  3985*  3988*
                    3993*  3994*  3995   4004*  4038   4053*
R4      =%000004     767#  1331*  1353   1384*  1387   1389   1395*  1397   1416*  1419   1421   1427*  1429
                    1608   1620   3548   3550*  3581*  3586*  3587   3592   3594*  3618   3620*  3922*  3923*
                    3930*  3931   3933   3970   3972*  3973*  3974*  3975   3976*  3990   3992*  4000*  4003*
                    4039   4052*
R5      =%000005     768#  1295*  1567   1581*  1582   1589*  1595*  1603*  1608*  1620*  1701*  1709   1731
                    1753   1775   1797   1819   1841   1863   1885   1910   1932   1957   1980   2002   2024
                    2049   2074   2096   2118   2140   2162   2184   2209   2231   2253   2315   2339   2356*
                    2374*  2419*  2429   2452   2468*  2473*  2478*  2488   2537   2607   2665   2723   2752
                    2774   2796   2818   2840   2927   2985   3055   3159   3219   3250   3281   3312   3362
                    3412   3901   3902*  3936   3939*  3971   3977*  3979*  3981*  3982*  3983*  3984   4002*
                    4040   4051*
R6      =%000006     769#   771
R7      =%000007     770#   772   1343*  1368*  1553*  1557*  1590*  1600*
SAVLOC   010126     3064#  3101
SEGVEC   000250      889#
SP      =%000006     771#   922*   923    924*   926    927    928*   931    933*   935*   941    1208*  1254*
                    1255*  1261   1264   1284   1287   1326*  1330*  1446   1459   1469   1567*  1568*  1588
                    1589   1594   1595   1599*  1601   1602   1603   1604   1723*  1745*  1767*  1789*  1811*
                    1833*  1855*  1877*  1902*  1924*  1949*  1972*  1994*  2016*  2041*  2066*  2088*  2110*
                    2132*  2154*  2176*  2201*  2223*  2245*  2267*  2333*  2339*  2344*  2354*  2356   2358
                    2378*  2452*  2458*  2467*  2468   2477*  2529*  2576*  2623*  2667*  2668*  2680*  2725*
                    2726*  2734*  2741*  2766*  2788*  2810*  2832*  2854*  2945*  3009*  3061*  3073   3094*
                    3095*  3100*  3164*  3176   3197*  3198*  3203*  3242*  3273*  3304*  3354*  3404*  3454*
                    3546*  3547*  3548*  3594   3595   3596   3602   3605   3620   3621   3622   3627   3630
                    3674*  3677   3679   3680   3706   3707   3710*  3720*  3721*  3722*  3723*  3724   3728
                    3749*  3750   3751   3752   3753   3767*  3768*  3771*  3772*  3781*  3786   3797   3798*
                    3799*  3800*  3824   3836*  3840*  3854*  3875   3880*  3901*  3902   3925*  3926*  3928
                    3936*  3939   3969*  3970*  3971*  3977   4002   4003   4004   4005*  4015*  4016   4035*
                    4036*  4037*  4038*  4039*  4040*  4041   4047*  4051   4052   4053   4054   4055   4056
                    4060*
SR0      000204      862#  1268   1269*  1282*  3924*  3938*
SR2      000206      863#
STACK = 001100       753#  1208   1723   1745   1767   1789   1811   1833   1855   1877   1902   1924   1949
                    1972   1994   2016   2041   2066   2088   2110   2132   2154   2176   2201   2223   2245
```

DCKBRE.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    2267    2741    2766    2788    2810    2832    2854    2945    3009    3094    3100    3197    3203
                    3242    3273    3304    3354    3404    3454
START   003620      1472    1610    1675#   3658
SWR   = 177570       759#    760    1271    1314    3646    3669*   3683    3685    3689    3696    3819    3827    3831
                    3834
SW0   = 000001       800#
SW00  = 000001       790#    800
SW01  = 000002       789#    799
SW02  = 000004       788#    798
SW03  = 000010       787#    797
SW04  = 000020       786#    796
SW05  = 000040       785#    795
SW06  = 000100       784#    794
SW07  = 000200       783#    793
SW08  = 000400       740     782#    792    3666    3683
SW09  = 001000       739     781#    791    1221    1222    3665    3681    3684    3687    3689    3707    3708    3712
                    3714    3811    3818    3834    3842
SW1   = 000002       799#
SW10  = 002000       737     780#   3639    3646    3659    3812    3819    3843
SW11  = 004000       736     779#   1219    1220    1221    3644    3664    3684    3687    3688    3694    3695    3696
                    3712    3714    3818    3842
SW12  = 010000       736     778#   3640    3647    3649    3659
SW13  = 020000       735     777#   3813    3827
SW14  = 040000       734     776#   3663    3669
SW15  = 100000       733     775#   3814    3831
SW2   = 000004       798#
SW3   = 000010       797#
SW4   = 000020       796#
SW5   = 000040       795#
SW6   = 000100       794#
SW7   = 000200       793#
SW8   = 000400       792#
SW9   = 001000       791#
T       007750      2930    2932    2940#
TBITVE= 000014       833#
TKB     012622      3771    3805#
TKS     012620      3769    3804#
TRAPVE= 000034       839#   1214#   1215*
TRTVEC= 000014       834#
TST1    003226      1502#
TST10   004046      1749#
TST11   004124      1771#
TST12   004202      1793#
TST13   004256      1815#
TST14   004332      1837#
TST15   004410      1859#
TST16   004464      1881#
TST17   004556      1906#
TST2    003250      1510#
TST20   004632      1928#
TST21   004726      1953#
TST22   005006      1976#
TST23   005062      1998#
TST24   005136      2020#
TST25   005230      2045#
TST26   005324      2070#
```

```
TST27   005400          2092#
TST3    003272          1518#
TST30   005456          2114#
TST31   005534          2136#
TST32   005610          2158#
TST33   005664          2180#
TST34   005762          2205#
TST35   006040          2227#
TST36   006116          2249#
TST37   006176          2311#
TST4    003334          1530#
TST40   006352          2425#
TST41   006534          2484#
TST42   006664          2533#
TST43   007014          2603#
TST44   007106          2661#
TST45   007170          2719#
TST46   007262          2748#
TST47   007340          2770#
TST5    003374          1545#
TST50   007414          2792#
TST51   007470          2814#
TST52   007546          2836#
TST53   007624          2858#
TST54   007674          2923#
TST55   007774          2981#
TST56   010076          3051#
TST57   010252          3155#
TST6    003716          1705#
TST60   010426          3213#
TST61   010534          3246#
TST62   010636          3277#
TST63   010740          3308#
TST64   011076          3358#
TST65   011234          3408#
TST66   011372          3458#
TST7    003772          1727#
TYPE  = 104400          1318    3648    3730    3745    3747    3789    3792    3796    3821    3830    3852    3863    3865
                        3869    3871    3876    3887    3996    4022#   4059
TYPERR  012776          3829    3852#
TO      007754          2928    2942#
USERTY  001630          1168#   1225#   1325#   1439    1452    1460    1606    3461
V       010052          2987    2992    3004#
VECSET  011450          1581    1701    2419    2478    3483#
V0      010056          2986    3006#
V1      010162          3056    3082#
V2      010166          3066    3086#
V3      010336          3160    3185#
V4      010342          3169    3189#
W       010514          3220    3239#
WHICH1  003376          1532    1547#   1699
XX      010616          3251    3270#
Y       010720          3282    3301#
YELLOW  010224          3080    3084    3092    3100#
Z0      011056          3313    3351#
Z1      011214          3363    3401#
```

DCKBRE.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
Z2      011352          3413    3451#
$ACCEP  012314          3719#   4029
$BDADR  001330           963#   3592*   3618*   4188
$BDDAT  001334           965#   3605*   3630*   3632*   4188
$BELL   012056          3648    3659#   3821    3843
$B20CT  013350          3882    3963#
$B201   013376          3965    3968#
$B2016  013362          3966#
$CM1  =  000013          968#    969#    970#    971#    972#    973#    974#    975#    976#    977#    978#    979#
$CM2  =  000026          968#    969#    970#    971#    972#    973#    974#    975#    976#    977#    978#    979#
$CM3  =  000013          966#    968
$CM4  =  000013          979#    980#    981#    982#    983#    984#    985#    986#    987#    988#    989#    990#
$CRLF   012453          3757#   3807    3830    3843    3852    3865    3871    3876
$DOAGN  012052          3650    3658#
$ENDAD  012042           854    3652    3654#
$EOP    011770          3473    3492    3642#
$ERFLG  001314           956#   1221*   3687    3693*   3714    3818*   3842
$ERRTB  001500          1031#   3861
$ERTTL  001312           955#   1222*   3822*   3842
$ESCAP  012774          1223*   3708*   3838    3840    3842#
$FILLS  001105           914#
$GDADR  001326           962#   3590*   3616*   4188
$GCDAT  001332           964#   1713*   1735*   1757*   1779*   1801*   1823*   1845*   1867*   1892*   1914*   1939*   1962*
                        1984*   2006*   2031*   2056*   2078*   2100*   2122*   2144*   2166*   2191*   2213*   2235*   2257*
                        2368*   2444*   2448*   2505*   2567*   2611*   2669*   2727*   2756*   2778*   2800*   2822*   2844*
                        2932*   2992*   2998*   3070*   3173*   3230*   3261*   3292*   3331*   3341*   3381*   3391*   3431*
                        3441*   3602    3627    3632    4188    4191
$HLT    012634          1212    3816#
$HLTAD  001324           961#   3824*   3825*   3826    3843    4185    4187    4188    4191
$ICNT   001304           952#   1219*   3700*   3701    3703*   3713
$ILLUP  013734          4033    4062#
$ITEMB  001322           959#   3826*   3843    3856
$KTOUT  013314          3926    3933#
$KT11   002304          1228*   1278#   1333    1356    1371    1531    1685    3566
$LF     012454          3747    3758#   3796    3807    3843
$LPADR  001306           953#   3691*   3706*   3710    3712
$LPERR  001310           954#   3691    3707*   3712    3836
$LSTBL  013346          3937*   3941#
$MXCNT  012312          3704    3713#
$NULL   001104           913#    933
$OCNT   013552          3968*   3997*   4009#
$OMODE  013554          3963*   3967*   3972    3975*   3986*   4011#
$OVER   012274          3670    3686    3692    3702    3709#
$PASS   001300           950#   1218*   3645*   3659    3698    3714
$POWER  013742          4059    4065#
$PWRDN  013606          1216    4033#   4057
$PWRUP  013654          4042    4047#
$QUES   012452          3745    3756#   3789    3807    3843
$RDSZ  =  000010        3774#
$READC  012456          3767#   4027
$READS  012512          3781#   4028
$REGAD  001336           966#   1291    1675    1679    3464*   3514*
$REGO   001340           968#   1322    3514
$REG1   001342           969#
$REG10  001360           976#
$REG11  001362           977#
```

```
$REG12  001364            978#
$REG2   001344            970#
$REG3   001346            971#
$REG4   001350            972#
$REG5   001352            973#
$REG6   001354            974#
$REG7   001356            975#
$SAVRE= ****** U         4029
$SAVR6  013740           4041*   4047    4048*   4049*   4064#
$SCOPE  012062           1209    3668#
$SETAD  001416            992#   1292    1687    2867    3467*   3522*   3568
$SETUP= 000017           1208#   1209    1212    1214    1216    3643
$SET0   001420            993#   1229*   3522
$SET1   001422            994#   1230*
$SET10  001440           1001#   1237*
$SET11  001442           1002#   1238*
$SET12  001444           1003#   1239*
$SET2   001424            995#   1231*
$SET3   001426            996#   1232*
$SET4   001430            997#   1233*
$SET5   001432            998#   1234*
$SET6   001434            999#   1235*
$SET7   001436           1000#   1236*
$SIZE   013124           1280    3901#
$SS   = 000001            727    1136#   1152#   1633#   1666#   2274#   2305#   2386#   2416#   2584#   2598#   2631#   2656#
                         2688#   2714#   2897#   2918#   2953#   2976#   3017#   3046#   3115#   3150#   3637#   3662#   3715#
                         3761#   3775#   3810#   3943#   4014#   4032#   4046#
$STUP = 177777           1208#
$SVLAD  012254           3678    3705#
$SWR  = 167400            724#    727     728     733     734     735     736     737     739     740     1219    1220    1221
                         1222    3639    3640    3644    3646    3647    3649    3659    3663    3664    3665    3666    3669
                         3681    3683    3684    3687    3688    3689    3694    3695    3696    3707    3708    3712    3714
                         3811    3812    3813    3814    3818    3819    3827    3831    3834    3842    3843
$TIMES  012310           1220*   3644*   3694*   3701    3704*   3712#
$TMPAD  001414            991#   1290    1691    3465*   3518*
$TMP0   001366            979#   3518
$TMP1   001370            980#
$TMP10  001406            987#
$TMP11  001410            988#
$TMP12  001412            989#
$TMP2   001372            981#
$TMP3   001374            982#
$TMP4   001376            983#
$TMP5   001400            984#
$TMP6   001402            985#
$TMP7   001404            986#
$TN   = 000001            727
$TPB    001102            912#    941*
$TPFLG  001106            915#    918
$TPS    001100            911#    939
$TRAP   013556           1214    4015#
$TRP  = 000010           4012#   4022    4027#   4028#   4029#   4030#
$TRPAD  013576           4019    4026#
$TSTNM  001302            951#   1211*   1680*   1688*   2882*   3643*   3685    3705*   3709    3714
$TTYIN  012624           3782    3783    3800    3806#
$TYPE   001110            918#   4022    4026
```

```
$XTSTR  012070          3672#
$OFILL  013553          3964*   3966*   3976    4010#
.     = 015152           845#    848     849#    853#    855#    858#    892#    943#    948#   1318    1321#   1505    1513
                        1521    1525    1538    1713    1719    1735    1741    1757    1763    1779    1785    1801    1807
                        1823    1829    1845    1851    1867    1873    1892    1898    1914    1920    1939    1945    1962
                        1968    1984    1990    2006    2012    2031    2037    2056    2062    2078    2084    2100    2106
                        2122    2128    2144    2150    2166    2172    2191    2197    2213    2219    2235    2241    2257
                        2263    2373    2472    2519    2524    2572    2611    2617    2674    2732    2756    2762    2778
                        2784    2800    2806    2822    2828    2844    2850    2941    3005    3070    3173    3238    3269
                        3300    3350    3400    3450    3659    3660    3712    3713    3714    3756    3757    3758    3759
                        3804    3805    3806#   3807    3842    3843    3926    4044    4063    4184#
```

```
COMMEN    840#  2269  2381  2579  2626  2683  2892  2948  3012  3110
ENDCOM    840#  2305  2416  2598  2656  2714  2918  2976  3046  3150
HLT       755#  1437  1450  1462  1506  1514  1522  1526  1541  1718  1722  1740  1744  1762  1766
         1784   1788  1806  1810  1828  1832  1850  1854  1872  1876  1897  1901  1919  1923  1944
         1948   1967  1971  1989  1993  2011  2015  2036  2040  2061  2065  2083  2087  2105  2109
         2127   2131  2149  2153  2171  2175  2196  2200  2218  2222  2240  2244  2262  2266  2372
         2377   2471  2476  2523  2528  2571  2575  2616  2622  2673  2679  2731  2740  2761  2765
         2783   2787  2805  2809  2827  2831  2849  2853  2940  2944  3004  3008  3078  3082  3090
         3099   3181  3185  3193  3202  3237  3241  3268  3272  3299  3303  3349  3353  3399  3403
         3449   3453
MORETA    728#   991
MULT      840#
POP       840#  3750  4051
PRINT     840#  1317
PUSH      840#  3720  4035
SCOPE     756#  1502  1510  1518  1530  1545  1552  1556  1705  1727  1749  1771  1793  1815  1837
         1859   1881  1906  1928  1953  1976  1998  2020  2045  2070  2092  2114  2136  2158  2180
         2205   2227  2249  2311  2425  2484  2533  2603  2661  2719  2748  2770  2792  2814  2836
         2858   2923  2981  3051  3155  3213  3246  3277  3308  3358  3408  3458  3642
SETTRA   4022#  4027  4028  4029
SPACE     840#
TYPBIN    840#
TYPDEC    840#
TYPNUM    840#
TYPOCT    840#  3879
TYPO16    840#
$$CMRE    944#   968   969   970   971   972   973   974   975   976   977   978
$$CMTM    944#   979   980   981   982   983   984   985   986   987   988   989
$$ESCA    840#
$$SCOP    840#
$$SET    4022#  4027  4028  4029
.ESCAP    840#
.SCOPE    728#   840#  1499  1507  1515  1527  1542  1702  1724  1746  1768  1790  1812  1834  1856
         1878   1903  1925  1950  1973  1995  2017  2042  2067  2089  2111  2133  2155  2177  2202
         2224   2246  2308  2422  2481  2530  2600  2658  2716  2745  2767  2789  2811  2833  2855
         2920   2978  3048  3152  3210  3243  3274  3305  3355  3405  3455
.SETUP    840#  1208
.SLASH    840#  2270  2305  2382  2416  2580  2598  2627  2656  2684  2714  2893  2918  2949  2976
         3013   3046  3111  3150
.STARS    840#  1134  1150  1629  1662  2270  2301  2382  2412  2580  2594  2627  2652  2684  2710
         2893   2914  2949  2972  3013  3042  3111  3146  3636  3661  3714  3760  3774  3809  3942
         4013   4031  4045
.$ACCE    724#  3714
.$ALSO    741#
.$CATC    724#   844
.$CMTA    724#   944
.$EOP     724#  3636
.$EQUA    724#   749
.$HEAD    724#
.$HLT     724#  3808
.$KT11    724#   859
.$OCTA    724#  3942
.$POWE    724#  4030
.$READ    724#  3760
.$SCOP    724#  3660
.$SETU    724#  1208
```

```
.$SWDO     724#    728
.$TERM    4022#
.$TRAP     724#   4012
.$TYPE     724#    890
```

```
ADD      928  1358  1361  1364  1378  1384  1410  1416  1443  1456  1465  1599  1619  1688  1961
        2352  2365  2465  2498  2502  2759  2882  3464  3465  3467  3470  3610  3612  3861  3926
        3930  3974
ASL     1272  3739  3740  3741  3858  3859  3860
BEQ      919  1316  1334  1377  1388  1404  1409  1420  1440  1453  1461  1513  1525  1538  1551
        1598  1607  1676  2342  2437  2456  2497  2545  2561  2872  2881  2591  3328  3378  3428
        3559  3604  3629  3650  3684  3686  3688  3690  3699  3732  3820  3823  3835  3839  3868
        3886  3991
BGE     3702
BGT     3734  3932  3998
BHI     2865  3580
BIC     1511  1523  1539  1587  1593  1717  1720  1739  1742  1761  1764  1783  1786  1805  1808
        1827  1830  1849  1852  1871  1874  1889  1896  1899  1918  1921  1935  1943  1946  1966
        1969  1988  1991  2010  2013  2028  2035  2038  2052  2060  2063  2082  2085  2104  2107
        2126  2129  2148  2151  2170  2173  2187  2195  2198  2217  2220  2239  2242  2261  2264
        2317  2355  2375  2435  2453  2474  2490  2514  2526  2542  2555  2573  2609  2618  2620
        2675  2677  2733  2738  2760  2763  2782  2785  2804  2807  2826  2829  2848  2851  2929
        2937  2942  3001  3006  3060  3086  3163  3189  3221  3229  3239  3252  3260  3270  3283
        3291  3301  3314  3344  3351  3364  3394  3401  3414  3444  3451  3459  3586  3742  3772
        3817  3988
BIS     1503  1519  1535  1547  1891  1937  2030  2054  2189  2346  2451  2508  2551  2614  2736
        2935  3226  3257  3288  3326  3376  3426  3484  3743  3993  3994
BISB    2847
BIT     1314  1504  1512  1520  1524  1537  1550  3646  3683  3689  3696  3819  3827  3834
BLOS    3784
BLT      936  3736  3738  3999
BMI     1276  1532  3670
BNE      925   932  1505  1521  2361  3089  3192  3462  3555  3589  3647  3652  3697  3788  3794
        3828  3874  3989  4050
BPL      940  1340  1357  1372  1686  3567  3770  3832  3987
BR       921   938  1259  1283  1319  1327  1363  1379  1393  1401  1411  1425  1433  1445  1677
        1719  1741  1763  1785  1807  1829  1851  1873  1898  1920  1945  1968  1990  2012  2037
        2062  2084  2106  2128  2150  2172  2197  2219  2241  2263  2345  2373  2447  2459  2472
        2501  2519  2524  2550  2572  2617  2672  2674  2730  2732  2762  2784  2806  2828  2850
        2874  2941  2997  3005  3080  3084  3092  3183  3187  3195  3238  3269  3300  3338  3350
        3388  3400  3438  3450  3564  3572  3615  3672  3678  3681  3692  3695  3744  3746  3790
        3888  3965  3980  4001  4044  4063
CLR     1211  1218  1219  1220  1222  1223  1225  1228  1229  1230  1231  1232  1233  1234  1235
        1236  1237  1238  1239  1240  1241  1242  1243  1244  1245  1246  1247  1248  1249  1250
        1251  1253  1265  1269  1281  1288  1335  1391  1399  1423  1431  1470  1622  1623  3057
        3102  3103  3105  3161  3205  3206  3208  3217  3499  3501  3510  3512  3549  3550  3560
        3643  3644  3694  3708  3725  3855  3912  3923  3938  3978  4048
CLRB    1221  2803  3693  3795
CMP     1261  1284  1339  1403  1446  1459  1486  1597  1601  1804  1826  1848  1870  1895  1917
        1942  1965  1987  2009  2034  2059  2081  2103  2125  2147  2169  2194  2859  2867  2879
        3578  3587  3602  3627  3651  3679  3701  3731  3733  3735  3783  3931
CMPB     931  3685  3787  3793
DEC     1277  1373  1405  2337  2343  2349  2433  2457  2462  2546  3737  3857
DECB     935  3986  3997
EMT      755
HALT     848   920  3833  4043  4062
INC     1262  1282  1325  1554  1596  2886  3645  3700  3705  3822  3924  3992  4000  4049
IOT      756
JMP      851  1341  1441  1454  1458  1464  1467  1472  1610  1624  1678  2216  2238  2260  2885
        3473  3492  3658
JSR      930   937  1280  1289  1343  1368  1571  1582  1693  1699  1709  1721  1731  1743  1753
```

```
                1765  1775  1787  1797  1809  1819  1831  1841  1853  1863  1875  1885  1900  1910  1922
                1932  1947  1957  1970  1980  1992  2002  2014  2024  2039  2049  2064  2074  2086  2096
                2108  2118  2130  2140  2152  2162  2174  2184  2199  2209  2221  2231  2243  2253  2265
                2315  2371  2376  2429  2470  2475  2488  2520  2527  2537  2570  2574  2607  2615  2621
                2665  2678  2723  2739  2752  2764  2774  2786  2796  2808  2818  2830  2840  2852  2927
                2943  2985  3007  3055  3098  3159  3201  3219  3236  3240  3250  3267  3271  3281  3298
                3302  3312  3348  3352  3362  3398  3402  3412  3448  3452  3472  3491  3654  3829  3882
MOV              922   923   927   933  1208  1209  1210  1212  1213  1214  1215  1216  1217  1254  1255
                1256  1257  1264  1266  1267  1271  1287  1290  1291  1292  1295  1326  1328  1330  1331
                1338  1350  1352  1353  1360  1362  1365  1367  1381  1389  1395  1397  1413  1421  1427
                1429  1435  1447  1469  1484  1487  1488  1498  1567  1568  1569  1573  1581  1584  1585
                1586  1588  1589  1594  1595  1602  1603  1608  1618  1620  1679  1687  1691  1695  1701
                1711  1712  1713  1716  1723  1733  1734  1735  1738  1745  1755  1756  1757  1760  1767
                1777  1778  1779  1782  1789  1799  1800  1801  1811  1821  1822  1823  1833  1843  1844
                1845  1855  1865  1866  1867  1877  1887  1888  1890  1892  1902  1912  1913  1914  1924
                1934  1936  1938  1939  1949  1959  1960  1962  1972  1982  1983  1984  1994  2004  2005
                2006  2016  2026  2027  2029  2031  2041  2051  2053  2055  2056  2066  2076  2077  2078
                2088  2098  2099  2100  2110  2120  2121  2122  2132  2142  2143  2144  2154  2164  2165
                2166  2176  2186  2188  2190  2191  2201  2211  2212  2213  2223  2233  2234  2235  2245
                2255  2256  2257  2267  2333  2335  2339  2344  2347  2354  2356  2358  2368  2378  2419
                2431  2440  2444  2448  2452  2458  2460  2467  2468  2477  2478  2491  2494  2495  2505
                2509  2515  2529  2539  2540  2548  2552  2556  2567  2576  2610  2611  2623  2667  2668
                2669  2680  2725  2726  2727  2734  2741  2754  2755  2756  2766  2776  2777  2778  2788
                2798  2799  2800  2810  2820  2821  2822  2832  2842  2843  2844  2854  2930  2932  2936
                2945  2987  2989  2992  2998  3009  3061  3062  3066  3069  3070  3094  3095  3100  3101
                3164  3165  3169  3172  3173  3197  3198  3203  3204  3222  3223  3227  3230  3233  3242
                3253  3254  3258  3261  3264  3273  3284  3285  3289  3292  3295  3304  3315  3318  3321
                3323  3331  3336  3339  3341  3345  3354  3365  3368  3371  3373  3381  3386  3389  3391
                3395  3404  3415  3418  3421  3423  3431  3436  3439  3441  3445  3454  3483  3514  3518
                3522  3526  3546  3547  3548  3568  3573  3575  3581  3590  3592  3594  3595  3596  3605
                3616  3618  3620  3621  3622  3630  3632  3649  3674  3675  3677  3680  3691  3703  3704
                3706  3707  3709  3710  3720  3721  3722  3723  3724  3726  3748  3749  3750  3751  3752
                3753  3767  3768  3781  3782  3797  3798  3799  3800  3824  3836  3840  3854  3862  3867
                3873  3875  3880  3901  3902  3903  3904  3905  3906  3907  3908  3909  3910  3911  3913
                3914  3915  3916  3917  3918  3919  3922  3925  3928  3933  3935  3936  3937  3939  3969
                3970  3971  3977  3984  4002  4003  4004  4005  4015  4016  4019  4033  4034  4035  4036
                4037  4038  4039  4040  4041  4042  4047  4051  4052  4053  4054  4055  4056  4057  4058
                4060
MOVB             924   941  3728  3729  3771  3786  3791  3818  3826  3856  3963  3964  3966  3967  3968
                3972  3975  3976  3995  4018
NEG             3973
NOP             3655  3656  3657
RESET           1536  3653
ROL             3669  3979  3981  3982  3983  3985
RTI              929  2676  2737  3711  3754  3773  3801  3837  3841  4061
RTS              942  1489  1553  1557  1590  1600  2374  2473  2619  3485  3503  3530  3606  3609  3631
                3634  3878  3940  4006  4020
SOB             2525
SPL             1258
SUB             1482  1680  2441  2564  2825  3316  3366  3416  3825  3934
TRAP            4022  4027  4028  4029
TST              926  1268  1333  1342  1356  1366  1371  1375  1387  1407  1419  1439  1452  1460  1531
                1604  1606  1612  1614  1675  1685  2340  2360  2436  2454  2496  2543  2560  2990  3073
                3087  3176  3190  3327  3377  3427  3461  3553  3557  3566  3607  3676  3698  3831  3838
                3885  3929  3990  4017
TSTB             918   939  1273  2781  3687  3769
```

```
.ASCII   3756  3757  4135  4143  4147  4150  4172
.ASCIZ   1321  3659  3758  3889  4065  4074  4078  4083  4087  4093  4097  4104  4110  4118  4126
         4139  4145  4149  4161  4178
.BLKB     943  3806
.BYTE     913   914   915   916   956   957   959   960  3755  3802  3803  3883  3884  4007  4008
         4009  4010
.ENABL    724
.END     4194
.ENDC     728   739   740   741   754   829   840   852   864   966   979   990  1018  1136  1152
         1208  1209  1212  1214  1216  1218  1224  1633  1666  2270  2274  2305  2382  2386  2416
         2580  2584  2598  2627  2631  2656  2684  2688  2714  2893  2897  2918  2949  2953  2976
         3013  3017  3046  3111  3115  3150  3637  3639  3640  3642  3643  3646  3648  3649  3659
         3660  3662  3669  3671  3682  3685  3687  3689  3696  3700  3705  3712  3714  3715  3761
         3774  3775  3782  3784  3789  3806  3807  3810  3816  3818  3824  3830  3831  3842  3843
         3884  3885  3943  4014  4019  4027  4028  4029  4030  4032  4041  4046  4051  4060  4061
         4068
.EQUIV    755   756   758   771   772   791   792   793   794   795   796   797   798   799   800
          819   820   821   822   823   824   825   826   827   828
.EVEN    1321  3890  4067  4184
.IF       728   737   739   740   741   753   801   829   848   864   966   979   990  1134  1150
         1208  1209  1212  1214  1216  1222  1629  1662  2270  2301  2305  2382  2412  2416  2580
         2594  2598  2627  2652  2656  2684  2710  2714  2893  2914  2918  2949  2972  2976  3013
         3042  3046  3111  3146  3150  3636  3639  3640  3641  3642  3646  3647  3649  3658  3659
         3661  3669  3681  3683  3684  3687  3688  3689  3696  3698  3712  3713  3714  3760  3774
         3782  3783  3788  3806  3809  3815  3817  3819  3827  3829  3830  3831  3834  3842  3843
         3883  3884  3942  4013  4019  4022  4027  4028  4029  4031  4041  4045  4051  4059  4060
         4065
.IFF      739   740   741   753   966  1134  1150  1208  1629  1662  2270  2301  2305  2382  2412
         2416  2580  2594  2598  2627  2652  2656  2684  2710  2714  2893  2914  2918  2949  2972
         2976  3013  3042  3046  3111  3146  3150  3637  3641  3643  3647  3659  3662  3682  3685
         3689  3713  3715  3761  3774  3775  3784  3788  3807  3810  3815  3819  3842  3883  3885
         3943  4014  4027  4028  4029  4032  4046  4060
.IFT     3695  3806  3830
.IFTF    3693  3802  3829
.IIF      724   725   726   727   733   734   735   736  1209  1219  1220  1221  3643  3644  3659
         3660  3663  3664  3665  3666  3694  3695  3707  3708  3712  3713  3714  3756  3757  3758
         3759  3804  3805  3807  3811  3812  3813  3814  3818  3842  3843  3881
.IRP      991  1208  3720  3750  3817  4035  4051
.LIST     724   728   840   848   890   944   966   968   969   970   971   972   973   974   975
          976   977   978   979   980   981   982   983   984   985   986   987   988   989   990
         1019  1100  1134  1135  1136  1150  1151  1152  1206  1208  1321  1503  1511  1519  1531
         1546  1629  1630  1631  1632  1633  1662  1663  1664  1665  1666  1706  1728  1750  1772
         1794  1816  1838  1860  1882  1907  1929  1954  1977  1999  2021  2046  2071  2093  2115
         2137  2159  2181  2206  2228  2250  2269  2270  2271  2272  2273  2274  2301  2302  2303
         2304  2305  2306  2312  2381  2382  2383  2384  2385  2386  2412  2413  2414  2415  2416
         2417  2426  2485  2534  2579  2580  2581  2582  2583  2584  2594  2595  2596  2597  2598
         2599  2604  2626  2627  2628  2629  2630  2631  2652  2653  2654  2655  2656  2657  2662
         2683  2684  2685  2686  2687  2688  2710  2711  2712  2713  2714  2715  2720  2749  2771
         2793  2815  2837  2859  2892  2893  2894  2895  2896  2897  2914  2915  2916  2917  2918
         2919  2924  2948  2949  2950  2951  2952  2953  2972  2973  2974  2975  2976  2977  2982
         3012  3013  3014  3015  3016  3017  3042  3043  3044  3045  3046  3047  3052  3110  3111
         3112  3113  3114  3115  3146  3147  3148  3149  3150  3151  3156  3214  3247  3278  3309
         3359  3409  3459  3636  3637  3660  3662  3714  3715  3760  3761  3774  3775  3808  3810
         3942  3943  4012  4014  4022  4027  4028  4029  4030  4032  4046
.MACRO    728   741   944  4022
.MCALL    724   840
```

```
.NLIST   724   728   840   848   890   944   966   968   969   970   971   972   973   974   975
         976   977   978   979   980   981   982   983   984   985   986   987   988   989   990
        1019  1100  1134  1135  1136  1150  1151  1152  1206  1208  1321  1503  1511  1519  1531
        1546  1629  1630  1631  1632  1633  1662  1663  1664  1665  1666  1706  1728  1750  1772
        1794  1816  1838  1860  1882  1907  1929  1954  1977  1999  2021  2046  2071  2093  2115
        2137  2159  2181  2206  2228  2250  2269  2270  2271  2272  2273  2274  2301  2302  2303
        2304  2305  2306  2312  2381  2382  2383  2384  2385  2386  2412  2413  2414  2415  2416
        2417  2426  2485  2534  2579  2580  2581  2582  2583  2584  2594  2595  2596  2597  2598
        2599  2604  2626  2627  2628  2629  2630  2631  2652  2653  2654  2655  2656  2657  2662
        2683  2684  2685  2686  2687  2688  2710  2711  2712  2713  2714  2715  2720  2749  2771
        2793  2815  2837  2859  2892  2893  2894  2895  2896  2897  2914  2915  2916  2917  2918
        2919  2924  2948  2949  2950  2951  2952  2953  2972  2973  2974  2975  2976  2977  2982
        3012  3013  3014  3015  3016  3017  3042  3043  3044  3045  3046  3047  3052  3110  3111
        3112  3113  3114  3115  3146  3147  3148  3149  3150  3151  3156  3214  3247  3278  3309
        3359  3409  3459  3636  3637  3660  3662  3714  3715  3760  3761  3774  3775  3808  3810
        3942  3943  4012  4014  4022  4027  4028  4029  4030  4032  4046
.PAGE    890   944  1019  1100  1152  1206  3636  3660  3714  3808  3891  4012  4030  4068
.REM       1
.REPT    848   968   979  1134  1150  1629  1662  2270  2301  2305  2382  2412  2416  2580  2594
        2598  2627  2652  2656  2684  2710  2714  2893  2914  2918  2949  2972  2976  3013  3042
        3046  3111  3146  3150
.SBTTL   890   944  1019  1100  1152  1206  3636  3660  3714  3760  3808  3942  4012  4030
.TITLE   724
.WORD    848   950   951   952   953   954   955   958   961   962   963   964   965   966   968
         969   970   971   972   973   974   975   976   977   978   979   980   981   982   983
         984   985   986   987   988   989   991   992   993   994   995   996   997   998   999
        1000  1001  1002  1003  1004  1005  1006  1007  1008  1009  1010  1011  1012  1013  1014
        1015  1016  3064  3167  3941  4185  4186  4187  4188  4191
```

```
ERRORS DETECTED:  0 HARD     22 SOFT
DEFAULT GLOBALS GENERATED:  0

*,DCKBRE/SOL/CRF/PAGNUM=DCKBRE
RUN-TIME: 31 32 5 SECONDS
RUN-TIME RATIO: 286/69=4.1
CORE USED:  15K  (29 PAGES)
```

Spooler runtime 15 Seconds, 66 KCS, 423 disk reads, 3 disk writes, 106 pages