

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBR-C-D
PRODUCT NAME: 11/40 - 11/45 CPU PARITY TEST
DATE CREATED: NOVEMBER 1973
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: BRUCE BURGESS

COPYRIGHT © 1973
DIGITAL EQUIPMENT CORPORATION

THE INFORMATION IN THIS STATEMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

ACTUAL DISTRIBUTION OF THE SOFTWARE DESCRIBED IN THIS DOCUMENT WILL BE SUBJECT TO TERMS AND CONDITIONS TO BE ANNOUNCED ON SOME FUTURE DATE BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE TO USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	OPERATOR ACTION
5.	OPERATIONAL SWITCH SETTINGS
5.1	SPECIAL USAGE
5.2	SPECIAL NOTE ON SW<12>
6.	SUBROUTINE ABSTRACTS
6.1	
THRU	SUBROUTINES EXPLAINED INDIVIDUALLY
6.15	
7.	ERROR PRINTOUTS
7.1	
THRU	ERROR PRINTOUT EXAMPLES AND EXPLANATIONS
7.10	
8.	RESTRICTIONS
9.	MISCELLANEOUS
9.1	EXECUTION TIME
9.2	PROGRAM TABLE LOCATIONS
9.3	PROGRAM TABLE SETUP WITH KT ₁₁ ENABLED
9.4	PROGRAM TABLE SETUP WITH KT ₁₁ DISABLED
9.5	STACK POINTER
9.6	MAINTENANCE HINT
10.	PROGRAM DESCRIPTION
10.1	PROGRAM FLOW DIAGRAM

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 SLPADR

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

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

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

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 FLAGCLR

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 SHLT

THIS ROUTINE CALLED (IN NUMEROUS PLACES THRU OUT THE PROGRAM) BY THE 'EMT' INSTRUCTION IS USED WHENEVER AN ERROR HAS BEEN DETECTED. THIS ROUTINE RELIES ON SWS<9,10,13,15> FOR FUNCTIONING AND IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. THE TYPEPR 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 'SLPADR' AS IT IS BEING ENTERED. 'LPADR' (IN THE COMMON TAG SECTION OF THE PROGRAM) MAY BE EXAMINED TO DETERMINE THE LAST TEST SUCCESSFULLY COMPLETED. THIS ROUTINE IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC,SML) AT ASSEMBLY TIME.

6.13 TRAPCATCHER

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

6.14 TYPERR

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

6.15 VECSET

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

7. ERROR PRINTOUTS

*** SPECIAL NOTE ***

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

*** END OF SPECIAL NOTE ***

7.1 HLT +1

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

7.2 HLT +2

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

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

7.3 HLT +3

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

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

- (1) THE EXPECTED HIGH ORDER ADDRESS BITS AND THE EXPECTED ABORT PC PUSHED ON THE STACK WERE BOTH WRONG.
 - A) IN THE CASE OF AN OLD MOS DESIGN WITH NO ADDRESS BITS ZEROS (0'S) WILL APPEAR UNDER THE ADDR, BITS COLUMNS.
- (2) THE EXPECTED HIGH ORDER ADDRESS BITS WERE CORRECT BUT THE WRONG ABORT PC WAS PUSHED ON THE STACK.
 - B) IN THIS CASE THE VALUES APPEARING UNDER THE ADDR, BITS COLUMNS WOULD BE THE SAME
- (3) THE EXPECTED HIGH ORDER ADDRESS BITS WERE 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 **

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 INFORMATION; PASS 3 BACK TO THE 1ST TABLE ENTRY INFORMATION, ETC.

THE ABSOLUTE CORE LOCATIONS FOR TABLE ENTRIES ARE AS FOLLOWS:

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

LOCATION 1342 UP TO 1364 WILL CONTAIN ANYMORE
REGISTERS FOUND

\$TMP2 (LOCATION 1366) WILL CONTAIN A PARITY MEMORY LOCATION
CORRESPONDING TO THE REGISTER IN \$REG0

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

\$SET2 (LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED
WITH THE CORRESPONDING VALUE IN \$TMP0

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

INTER0 (LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED
WITH THE PARITY REGISTER IN \$REG0

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

9.3 PROGRAM TABLE SET UP WITH KT11 ENABLED

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

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

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

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

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

```

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

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

- (3) THE PHYSICAL ADDRESS VALUE FROM ABOVE IS THE VALUE USED BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5) WHICH WILL DROP THE PHYSICAL ADDRESS DOWN SO AS NOT TO DESTROY THE .ABS LOADER (I.E. - 376 IS SUBTRACTED) THUS GIVING A PHYSICAL ADDRESS FOR THE 2 LOCATION MAP CREATION AND TESTING. THIS VALUE IS ALWAYS PRESENT IN R1. (GENERAL PURPOSE REGISTER 1)
- (4) THE ZERO IN THE LAST REGISTER COLUMN LOCATION IS THE PROGRAM TABLE TERMINATION INDICATOR
- (5) A '1' IN THE ILEAVE COLUMN MEANS NO INTERLEAVING
A '2' IN THE ILEAVE COLUMN MEANS 2-WAY INTERLEAVING

.
.
ETC. (UP TO 8-WAY)

9.4 PROGRAM TABLE SETUP WITH KT11 DISABLED

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

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

LOC.	REGISTER COLUMN	LOC.	MEMORY COLUMN	LOC.	OFFSET COLUMN	ILEAVE COLUMN
1340	172100	1366	17700	1420	0	1
1342	172102	1370	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 SM₀,DM₀' INSTRUCTION
- (D) THE 'POP' ON AN RTS INSTRUCTION
- (E) 1ST AND 2ND 'POP' ON AN RTI INSTRUCTION
- (F) PS AN PC FETCH INSTRUCTIONS
- (G) INDEXED WORD INSTRUCTIONS (DM6,DM7,SM6)
- (H) CONDITIONAL BRANCH NOT OK INSTRUCTIONS
- (I) STACK VIOLATIONS IN 'RED' AND 'YELLOW' ZONES

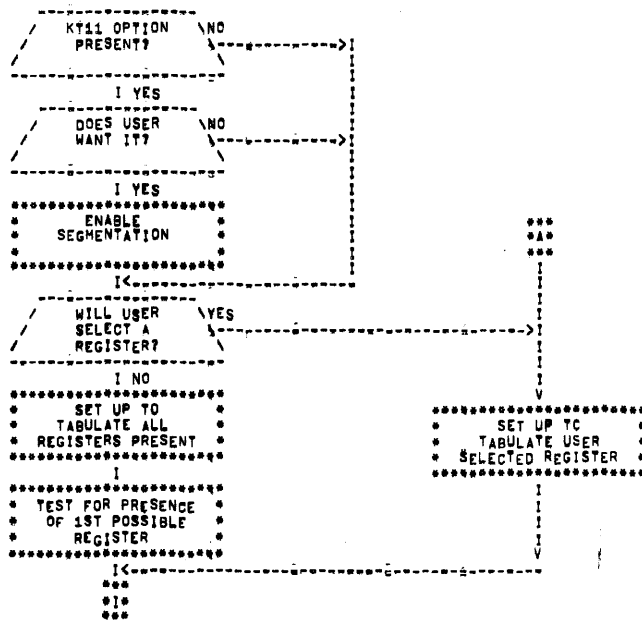
THIS PROGRAM USED IN CONJUNCTION WITH MAINDEC-11-DCMFA SHOULD PROVIDE A PRETTY THOROUGH TEST OF PARITY.

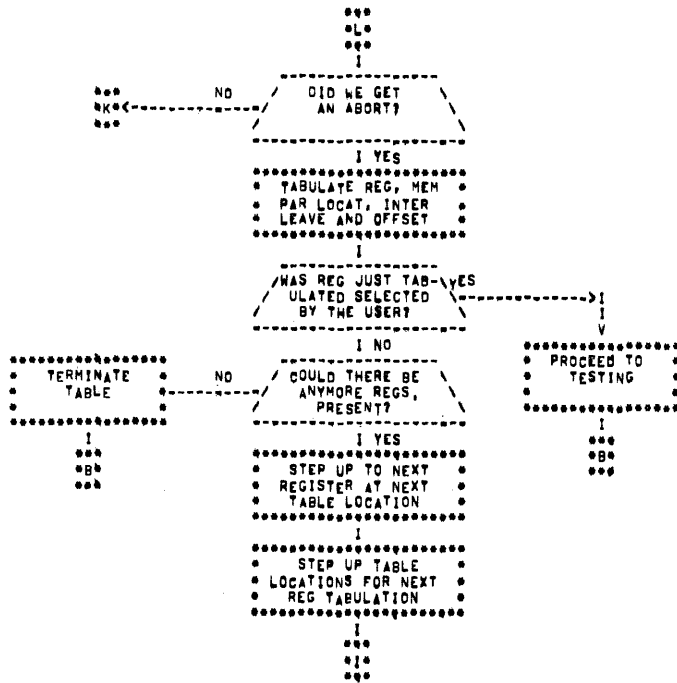
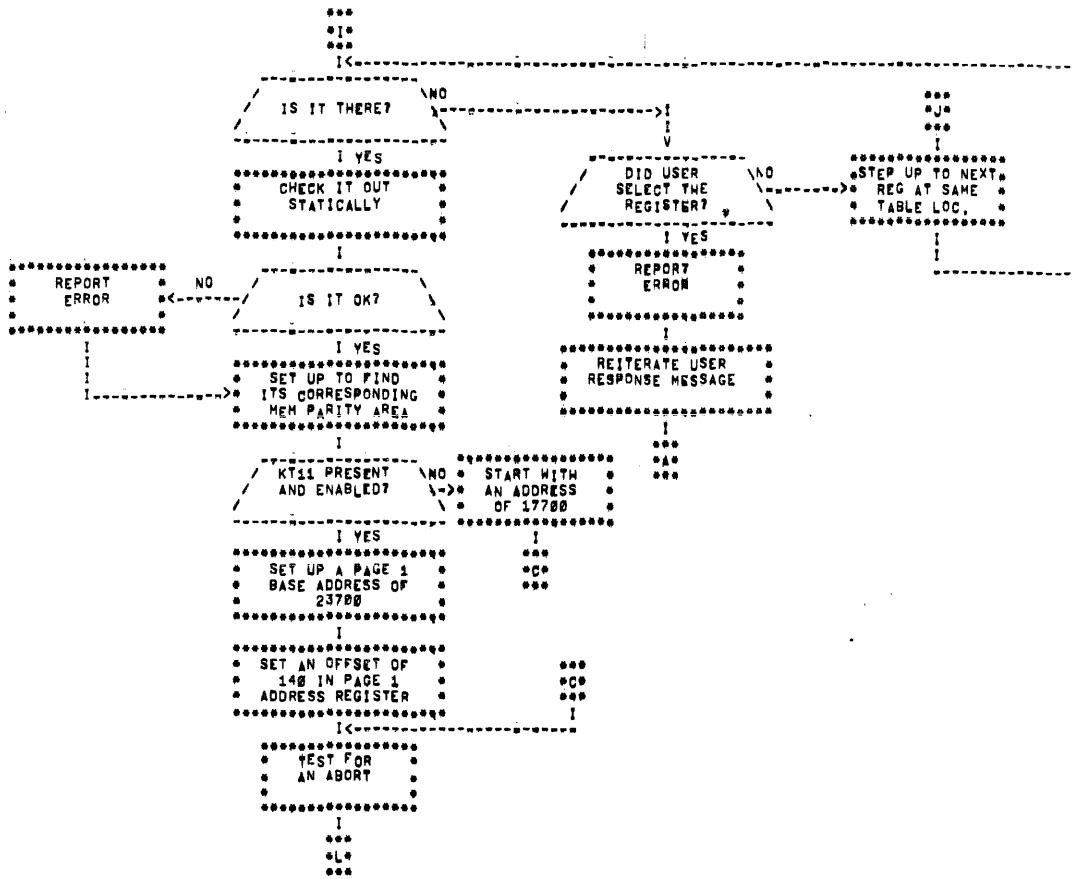
10.1 PROGRAM FLOW DIAGRAM

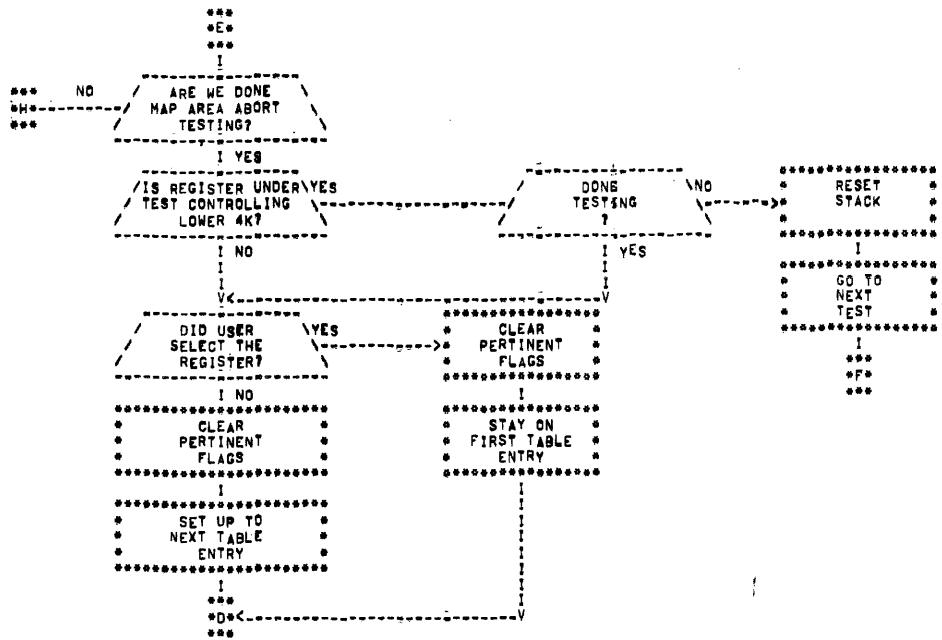
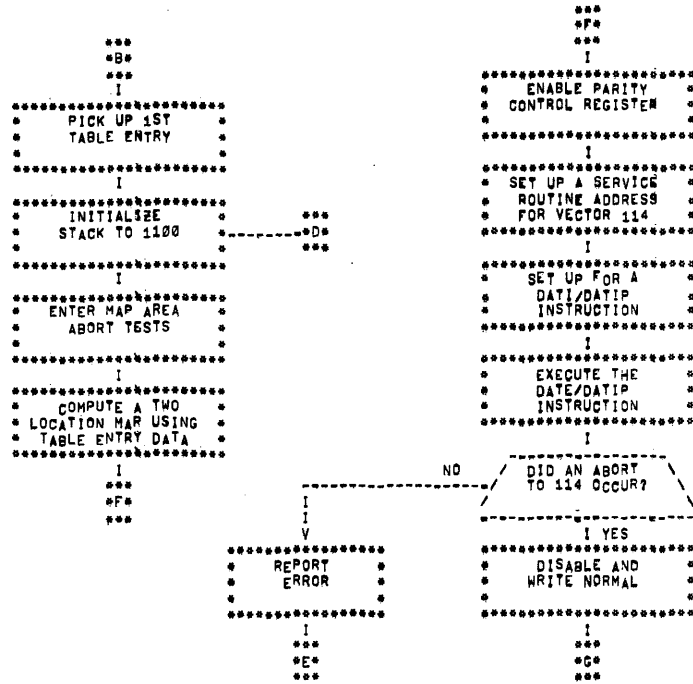
FLOW CHART

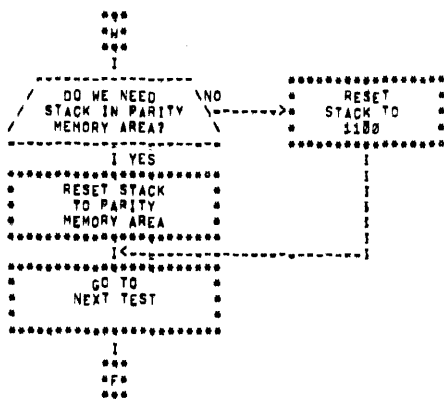
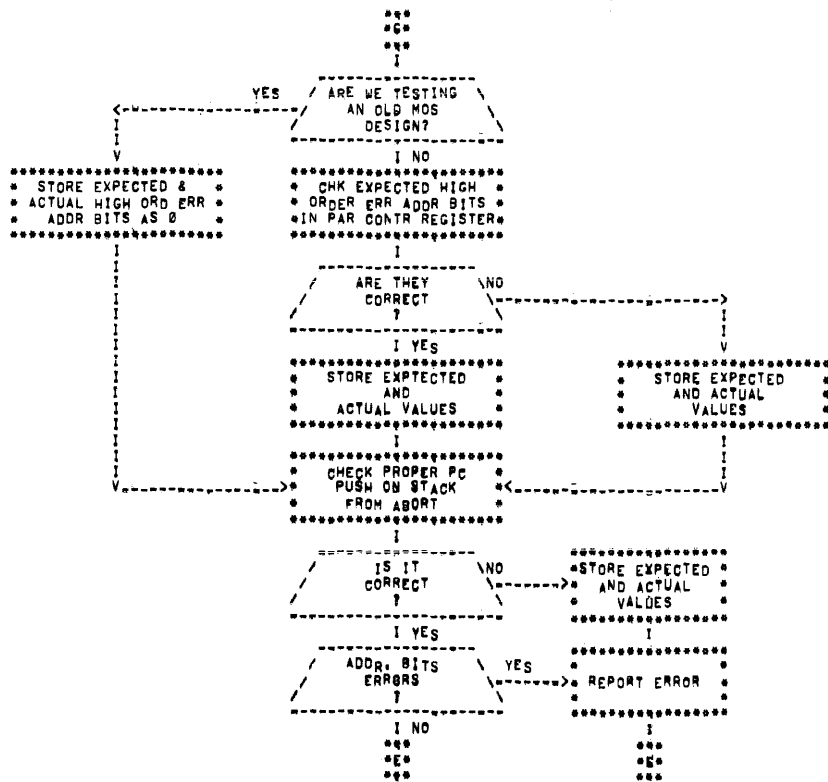
 CPU PARITY FLOW DIAGRAM

COPYRIGHT 1974
 DIGITAL EQUIPMENT CORPORATION
 MAYNARD, MASS. 01754









1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

.TITLE MAINDEC-11-DCKBR-C
;COPYRIGHT 1973 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01794
;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 KEY 11 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
STACK# 1100
;#####
;EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
;EQUIV IOT,SCOPE ;BASIC DEFINITION OF SCORE CALL
PS# 177776 ;PROCESSOR STATUS WORD
;EQUIV PS,PSW
SWR# 177570 ;SWITCH REGISTER
DISPLAY#SWR
;
;REGISTER DEFINITION
R0# X0 ;GENERAL REGISTER
R1# X1 ;GENERAL REGISTER
R2# X2 ;GENERAL REGISTER
R3# X3 ;GENERAL REGISTER
R4# X4 ;GENERAL REGISTER
R5# X5 ;GENERAL REGISTER
R6# X6 ;GENERAL REGISTER
R7# X7 ;GENERAL REGISTER
;EQUIV R6,SP ;STACK POINTER
;EQUIV R7,PC ;PROGRAM COUNTER
;
;SWITCH DEFINITION
SW15# 100000

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

SW14# 40000
SW13# 20000
SW12# 10000
SW11# 4000
SW10# 2000
SW09# 1000
SW08# 400
SW07# 200
SW06# 100
SW05# 40
SW04# 20
SW03# 10
SW02# 4
SW01# 2
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
BIT15# 100000
BIT14# 40000
BIT13# 20000
BIT12# 10000
BIT11# 4000
BIT10# 2000
BIT09# 1000
BIT08# 400
BIT07# 200
BIT06# 100
BIT05# 40
BIT04# 20
BIT03# 10
BIT02# 4
BIT01# 2
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

```

109 ;VECTOR ADDRESSES
110 000004 ERRVEC= 4
111 000010 RESVEC= 10
112 000014 TBIVVEC=14
113 000014 TRTVEC= 14
114 000014 BPTVEC= 14
115 000020 IOTVEC= 20
116 000024 PWRVEC= 24
117 000030 EMTVEC= 30
118 000034 TRAPVEC=34
119 000001 N=1
120
121 ;DEFINE STARTING INI FOR, SCOPE
122 ;ROUTINE WHICH PRINTS TEST NO.
123 ;AND APPROPRIATE COMMENT
124
125 ;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
126 ;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS
127
128 ;=200
129
130 000200 000137 001706 JMP *#BEGIN ;JUMP TO STARTING ADDRESS OF PROGRAM
131
132 ;KT11-D STATUS REGISTER ADDRESSES
133
134 000204 177572 SR0; 177572
135 000206 177576 SR2; 177576
136
137 ;KERNAL PAGE DESCRIPTOR REGISTERS
138
139 000210 172300 KPDR0; 172300
140 000212 172302 KPDR1; 172302
141 000214 172304 KPDR2; 172304
142 000216 172306 KPDR3; 172306
143 000220 172310 KPDR4; 172310
144 000222 172312 KPDR5; 172312
145 000224 172314 KPDR6; 172314
146 000226 172316 KPDR7; 172316
147
148 ;KERNAL PAGE ADDRESS REGISTERS
149
150 000230 172340 KPAR0; 172340
151 000232 172342 KPAR1; 172342
152 000234 172344 KPAR2; 172344
153 000236 172346 KPAR3; 172346
154 000240 172350 KPAR4; 172350
155 000242 172352 KPAR5; 172352
156 000244 172354 KPAR6; 172354
157 000246 172356 KPAR7; 172356
158
159 ;KT11 VECTOR ADDRESS
160
161 000250 000250 000252 SEGVEC; 250,252

```

```

162
163
164 001100 ;*****
165 ;=1100
166
167 ;ROUTINE TO TYPE ASCII MESSAGE, MESSAGE MUST TERMINATE WITH A 0 BYTE.
168 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
169 ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
170 ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
171
172 ;CALL;
173 ;1) USING A TRAP INSTRUCTION
174 ; TYPE ,MESADR ;MESADR IS FIRST ADDRESS OF AN ASCII STRING
175 ;
176 ; TYPE
177 ; MESADR
178
179 ;2) USING A JSR INSTRUCTION
180 ; MOV PS,*(SP) ;PUSH PROCESSOR STATUS WORD ON THE STACK
181 ; JSR PC,TYPE ;CALL TYPE ROUTINE
182 ; MESADDR ;FIRST ADDRESS OF MESSAGE
183
184 001100 177564 STPS; 177564 ;TTY PRINTER STATUS REG. ADDRESS
185 001102 177566 STPB; 177566 ;TTY PRINTER BUFFER REG. ADDRESS
186 001104 000 $NULL; ,BYTE 0 ;CONTAINS NULL CHARACTER FOR FILLS
187 001106 000 STPFLG; ,BYTE 2 ;CONTAINS # OF FILLER CHARACTERS REQUIRED
188 001107 000 STPFLG; ,BYTE 0 ;"TERMINAL AVAILABLE" FLAG (0=YES)
189 ;RESERVED
190
191 001110 105767 177772 STYPE; TSTB STPFLG
192 001114 001402 BEQ 65 ;IS THERE A TERMINAL?
193 001116 000000 HALT ;IF YES
194 001120 000407 BR 75 ;HALT HERE IF NO TERMINAL
195 001122 010046 MOV RB,*(SP) ;I LEAVE
196 001124 017600 000002 65; MOV RB,*(SP),RB ;SAVE RB
197 001130 112046 1$; MOV RB,(RB),*(SP) ;GET ADDRESS OF ASCII STRING
198 001132 001005 BNE 25 ;PUSH CHARACTER TO BE TYPED ONTO STACK
199 001134 005726 TST (SP)+ ;IF TERMINATOR POP IT OFF THE STACK
200 001136 012600 MOV (SP)+,RB ;IF TERMINATOR
201 001140 062716 000002 7$; ADD #2,(SP) ;RESTORE RB
202 001144 000002 RTI ;ADJUST RETURN PC
203 001146 004767 000026 2$; JSR PC,5$ ;RETURN
204 001152 122726 000012 3$; CHPB #12,(SP)+ ;GO TYPE THIS CHARACTER
205 001156 001364 BNE 15 ;CHECK IF THE CHAR. TYPED WAS A LINE FEED
206 001160 010746 177720 MOV $NULL,*(SP)+ ;GO GET NEXT CHAR. IF NOT LINE FEED
207 ;AND THE NULL CHAR.
208 001164 105366 000001 4$; DECIB 1,(SP) ;GET # OF FILLER CHARS, NEEDED
209 001170 002770 BLT 35 ;DOES A NULL NEED TO BE TYPED?
210 001172 004767 000002 JSR PC,5$ ;IF YES
211 001176 000772 BR 45 ;IF NO--DO POP THE NULL OFF OF STACK
212 001200 105777 177674 5$; TSTB *STPS ;GO TYPE A NULL
213 001204 100375 BPL 35 ;CHECK IF THE CHAR. TYPED WAS A LINE FEED
214 001206 116677 000002 177666 MOVB 2(SP),*STPB ;IF NOT LINE FEED
215 001214 000207 RTS PC ;GO TYPE A NULL
216 001216 000002 ,BLKB 62 ;LOOP
;WAIT UNTIL PRINTER IS READY
;LOAD CHAR TO BE TYPED INTO DATA REG.
;RESERVE SOME MORE CORE FOR OVERLAY CAPABILITIES

```

```

216 ;*****
217 ;COMMON TAGS
218
219
220      001300      ,#1300
221
222 001300 000000 SPASSI ,WORD 0 ICONTAINS PASS COUNT
223 001302 000000 STSTNMI ,WORD 0 ICONTAINS THE TEST NUMBER
224 001304 000000 SIONTI ,WORD 0 ICONTAINS SUBTEST ITERATION COUNT
225 001306 000000 SLPADRI ,WORD 0 ICONTAINS SCOPE LOOP ADDRESS
226 001310 000000 SLPERRI ,WORD 0 ICONTAINS SCOPE RETURN FOR ERRORS
227 001312 000000 SERTLTI ,WORD 0 ICONTAINS TOTAL ERRORS DETECTED
228 001314 000 SERFLGI ,BYTE 0 ICONTAINS ERROR FLAG
229 001315 000 ,BYTE 0 IRESERVED--NOT TO BE USED
230 001316 000000 000000 ,WORD 0,2 IRESERVED--NOT TO BE USED
231 001322 000 SITEMBI ,BYTE 0 ICONTAINS ITEM CONTROL BYTE
232 001323 000 ,BYTE 0 IRESERVED--NOT TO BE USED
233 001324 000000 SHLTADI ,WORD 0 ICONTAINS PC OF LAST HLT INSTRUCTION
234 001326 000000 SGGADRI ,WORD 0 ICONTAINS ADDRESS OF 'GOOD' DATA
235 001330 000000 SBDADRI ,WORD 0 ICONTAINS ADDRESS OF 'BAD' DATA
236 001332 000000 SGGDATI ,WORD 0 ICONTAINS 'GOOD' DATA
237 001334 000000 SBDAT1I ,WORD 0 ICONTAINS 'BAD' DATA
238 001336 000000 SREGADI ,WORD 0 ICONTAINS THE ADDRESS FROM
239 WHICH ($REG0) WAS OBTAINED
240 001340 000000 SREG0I ,WORD 0 ICONTAINS (($REG0)+0)
241 001342 000000 SREG1I ,WORD 0 ICONTAINS (($REG0)+2)
242 001344 000000 SREG2I ,WORD 0 ICONTAINS (($REG0)+4)
243 001346 000000 SREG3I ,WORD 0 ICONTAINS (($REG0)+6)
244 001350 000000 SREG4I ,WORD 0 ICONTAINS (($REG0)+10)
245 001352 000000 SREG5I ,WORD 0 ICONTAINS (($REG0)+12)
246 001354 000000 SREG6I ,WORD 0 ICONTAINS (($REG0)+14)
247 001356 000000 SREG7I ,WORD 0 ICONTAINS (($REG0)+16)
248 001360 000000 SREG10I ,WORD 0 ICONTAINS (($REG0)+20)
249 001362 000000 SREG11I ,WORD 0 ICONTAINS (($REG0)+22)
250 001364 000000 SREG12I ,WORD 0 ICONTAINS (($REG0)+24)
251 001366 000000 STMP0I ,WORD 0 USER DEFINED
252 001370 000000 STMP1I ,WORD 0 USER DEFINED
253 001372 000000 STMP2I ,WORD 0 USER DEFINED
254 001374 000000 STMP3I ,WORD 0 USER DEFINED
255 001376 000000 STMP4I ,WORD 0 USER DEFINED
256 001400 000000 STMP5I ,WORD 0 USER DEFINED
257 001402 000000 STMP6I ,WORD 0 USER DEFINED
258 001404 000000 STMP7I ,WORD 0 USER DEFINED
259 001406 000000 STMP10I ,WORD 0 USER DEFINED
260 001410 000000 STMP11I ,WORD 0 USER DEFINED
261 001412 000000 STMP12I ,WORD 0 USER DEFINED
262 ;THE FOLLOWING TAG(S) ARE USER DEFINED
263 001414 000000 STMPADI ,WORD 0
264 001416 000000 SSETADI ,WORD 0
265 001420 000000 SSET0I ,WORD 0 ITHESE LOCATIONS CONTAIN THE
266 001422 000000 SSET1I ,WORD 0 APPROPRIATE OFFSET VALUES
267 001424 000000 SSET2I ,WORD 0 IFOR THE PARITY CONTROL
268 001426 000000 SSET3I ,WORD 0 REGISTERS WHEN MEMORY
269 001430 000000 SSET4I ,WORD 0 MANAGEMENT IS ENABLED

```

```

270 001432 000000 SSET5I ,WORD 0 IDURING PROGRAM EXECUTION
271 001434 000000 SSET6I ,WORD 0
272 001436 000000 SSET7I ,WORD 0
273 001440 000000 SSET10I ,WORD 0
274 001442 000000 SSET11I ,WORD 0
275 001444 000000 SSET12I ,WORD 0
276 001446 000000 NTERADI ,WORD 0
277 001450 000000 NTER0I ,WORD 0 ITHESE LOCATIONS CONTAIN THE
278 001452 000000 NTER1I ,WORD 0 APPROPRIATE INTERLEAVE FACTORS
279 001454 000000 NTER2I ,WORD 0 IFOR THE PARITY CONTROL
280 001456 000000 NTER3I ,WORD 0 REGISTERS (IF ANY)
281 001460 000000 NTER4I ,WORD 0
282 001462 000000 NTER5I ,WORD 0
283 001464 000000 NTER6I ,WORD 0
284 001466 000000 NTER7I ,WORD 0
285 001470 000000 NTER10I ,WORD 0
286 001472 000000 NTER11I ,WORD 0
287 001474 000000 NTER12I ,WORD 0
288 001476 000000 NEWSTKI ,WORD 0
289 IEND OF USER DEFINED TAG(S)
290

```

```

;THE FOLLOWING TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;THIS INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;LOCATION ITEMS, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;NOTE1: IF ITEM# IS 0 THE ONLY PERTINENT DATA IS (SHLTAD).
;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
;
;ERRRBI
;NOTE: ALL NUMBERS ARE TYPED AS 6-DIGIT OCTAL NUMBERS
;ITEM 1
EM1      ITEST DIDN'T ABORT
DH5      IPROGRAM PC
          IREGISTER UNDER TEST
          IEXPECTED ABORT PC
          ISHLTAB,PARITY,$GDDAT
DT5      0
;ITEM 2
EM2      IFATAL ERROR TO PROGRAM
DH1      IPROGRAM PC
          IREGISTER UNDER TEST
          IEXPECTED ABORT PC
          ISHLTAB, PARITY
DT1      0
;ITEM 3
EM3      IABORTED INCORRECTLY
DH4      IPROGRAM PC
          IREGISTER UNDER TEST
          IEXPECTED BITS 5 THRU 11
          IACTUAL BITS 5 THRU 11
          IEXPECTED ABORT PC
          IACTUAL ABORT PC
          ISHLTAB,PARITY,$GDADR,$BDADR,$GDDAT,$BDDAT
DT4      0
;ITEM 4
EM4      INO PARITY MEMORY FOUND BELOW 28K
DH2      IREGISTER UNDER TEST
DT2      0
;ITEM 5
EM5      IRESET DOESN'T WORK
DH1      IPROGRAM PC
          IREGISTER UNDER TEST
          ISHLTAB, PARITY
DT1      0
;ITEM 6
EM6      IUSER SELECTED REGISTER NOT PRESENT
DH3      IPROGRAM PC
DT3      0
          0
;ITEM 7
EM7      INO PARITY MEMORY FOUND AT ALL
DH2      IREGISTER UNDER TEST
DT2      0
          0
;ITEM 10
EM10     IDIDN'T ABORT OR RECOGNIZE
          ISTACK VIOLATION
          IPROGRAM PC
          IREGISTER UNDER TEST
          IEXPECTED ABORT PC
          ISHLTAB, PARITY, $GDDAT
DT5      0
;ITEM 11
EM11     IABORTED BUT STACK VIOLATION
          INOT RECOGNIZED
          IPROGRAM PC
          IREGISTER UNDER TEST
          ISHLTAB, PARITY
DT1      0
;ITEM 12
EM12     ISTACK VIOLATION PICKED UP BUT
          IABORT NOT RECOGNIZED
          IPROGRAM PC
          IREGISTER UNDER TEST
          ISHLTAB, PARITY
DT1      0
          0

```

```

;ITEM 7
EM7      INO PARITY MEMORY FOUND AT ALL
DH2      IREGISTER UNDER TEST
DT2      0
          0
;ITEM 10
EM10     IDIDN'T ABORT OR RECOGNIZE
          ISTACK VIOLATION
          IPROGRAM PC
          IREGISTER UNDER TEST
          IEXPECTED ABORT PC
          ISHLTAB, PARITY, $GDDAT
DT5      0
;ITEM 11
EM11     IABORTED BUT STACK VIOLATION
          INOT RECOGNIZED
          IPROGRAM PC
          IREGISTER UNDER TEST
          ISHLTAB, PARITY
DT1      0
;ITEM 12
EM12     ISTACK VIOLATION PICKED UP BUT
          IABORT NOT RECOGNIZED
          IPROGRAM PC
          IREGISTER UNDER TEST
          ISHLTAB, PARITY
DT1      0
          0

```

```

372
373
374
375 ;SYSTEM PARITY REGISTER NOTES FOR MF11 AND MS11
376
377
378
379 ;BIT ASSIGNMENTS FOR THE MF11 PARITY REGISTER IS AS FOLLOWS
380
381 ;BIT15      PARITY ERROR
382 ;BITS 11-5  ERROR ADDRESS           HIGH ORDER ADDRESS BITS
383 ;           ;                       ;OP ADDRESS OF MOST RECENT ERROR
384 ;           ;                       ;(BITS 17 THRU 11)
385 ;BIT02      WRITE                     ;NORMAL PARITY (ODD) WHEN CLEAR
386 ;           ;                       ;OTHER PARITY (EVEN) WHEN SET
387 ;BIT00      ERROR ACTION ENABLE       ;NO ACTION WHEN CLEAR
388 ;           ;                       ;TRAP TO VECTOR 114 WHEN SET
389
390 ;NOTE: THE ABOVE BITS ARE READ/WRITE AND CAN BE CLEARED BY 'INIT' (EXCEPT BITS 11-5)
391
392
393 ;BIT ASSIGNMENTS FOR THE MS11 PARITY REGISTER IS AS FOLLOWS
394
395 ;BIT15      PARITY ERROR
396 ;BIT02      WRITE                     ;NORMAL PARITY (EVEN) WHEN CLEAR
397 ;           ;                       ;OTHER PARITY (ODD) WHEN SET
398 ;BIT00      ERROR ACTION ENABLE       ;NO ACTION WHEN CLEAR
399 ;           ;                       ;TRAP TO VECTOR 114 WHEN SET
400
401
402 ;NOTE: THERE ARE NO ERROR ADDRESS BITS IN THE CURRENT MS11 PARITY REGISTER
403 ; HOWEVER, THERE WILL BE IN A LATER VERSION WHICH WILL BE
404 ; HANDLED PROPERLY BY THIS PROGRAM
405
406
407
408
409 ;SPECIAL NOTE---THERE ARE 2 GENERAL PURPOSE REGISTERS USED IN THE
410 ; PROGRAM FOR SPECIFIC CIRCUMSTANCES. THEY ARE:
411
412
413 ; R1 - WILL ALWAYS CONTAIN THE 1ST ADDRESS OF THE 2
414 ; LOCATION MAP USED FOR TESTING,
415 ; THE CONTENTS OF R1 IS DETERMINED BY THE 'COMPUT'
416 ; ROUTINE SHOWN FURTHER DOWN.
417 ; EXAMINATION OF R1 WILL TELL YOU WHERE IN PARITY
418 ; MEMORY TESTING IS BEING CONDUCTED.
419
420
421 ; R5 - WILL ALWAYS CONTAIN THE ADDRESS OF THE ROUTINE
422 ; FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS.
423

```

```

424
425
426
427 ;*****
428 ;
429 ;MISCELLANEOUS COMMON PARITY VARIABLES AND FLAGS
430 ;
431 ;*****
432 001620 000114 INTVEC1 114 ;PARITY INTERRUPT VECTOR ADDRESS
433 001622 000000 PARITY1 0 ;CONTAINS PARITY REGISTER IN USE
434 001624 000000 PSPCORONES1 0 ;FLAG TO INDICATE TO 'CHECKLOC'
435 ;ROUTINE THAT A PS OR PC FETCH
436 ;OR A NONE ABORT WAS DONE
437 ;0 = NO, 1 = YES
438 001626 000000 MSREGFLAG1 0 ;INDICATES PARITY TYPE
439 ;0 = CORE, 1 = MOS
440 001630 000000 USERTYPE1 0 ;INDICATES USER SELECTION OF
441 ;PARITY REGISTER
442 ;0 = PROGRAM FIND
443 ;1 = USER SELECTION
444 001632 000000 BLKCNT1 0 ;CONTAINS THE NUMBER OF CONSEC-
445 ;UTIVE LOCATIONS TO BE TESTED
446 ;DURING PROGRAM TABULATION TO
447 ;COVER CASES OF MEMORY INTER-
448 ;LEAVING
449 001634 000000 RESTOREBASE1 0 ;HOLDS PAGE 1 ADDRESS
450 ;OR CURRENT MEMORY ADDRESS FOR
451 ;RESTORATION DURING
452 ;RUNNING OF PROGRAM. IT IS USED IF
453 ;WE HAVE CHECKED CONSECUTIVE
454 ;LOCATIONS WITHOUT AN ABORT BEFORE
455 ;GOING TO NEXT OFFSET WHICH WILL
456 ;PUT US IN ANOTHER BANK
457 001636 000000 LEAFCNT1 0 ;CONTAINS THE NO. OF ABORTS
458 ;ENCOUNTERED IN DETERMINATION OF
459 ;AN INTERLEAVE FACTOR
460 001640 000000 MEMADI 0 ;CONTAINS A BASE ADDRESS OR A
461 ;CURRENT MEMORY ADDRESS USED IN
462 001642 000000 CPU40i 0 ;FLAG TO INDICATE PROCESSOR
463 ;0 = 11/45, 1 = 11/40
464 ;PARITY TABLE CREATION
465
466 ;THE FOLLOWING TABLE IS USED TO DETERMINE THE
467 ;INTERLEAVE FACTOR FOR THE CONTROL REGISTERS
468 ;INTERTABLE1
469 001644 000003 000001 3,.1 ;3 ABORTS ON 3 CONSECUTIVE LOCS. = 1 WAY LEAVE
470 001650 000006 000002 6,.2 ;3 ABORTS ON 6 CONSECUTIVE LOCS. = 2 WAY LEAVE
471 001654 000011 000003 9,.3 ;3 ABORTS ON 9 CONSECUTIVE LOCS. = 3 WAY LEAVE
472 001660 000014 000004 12,.4 ;3 ABORTS ON 12 CONSECUTIVE LOCS. = 4 WAY LEAVE
473 001664 000017 000005 15,.5 ;3 ABORTS ON 15 CONSECUTIVE LOCS. = 5 WAY LEAVE
474 001670 000022 000006 18,.6 ;3 ABORTS ON 18 CONSECUTIVE LOCS. = 6 WAY LEAVE
475 001674 000025 000007 21,.7 ;3 ABORTS ON 21 CONSECUTIVE LOCS. = 7 WAY LEAVE
476 001700 000030 000010 24,.8 ;3 ABORTS ON 24 CONSECUTIVE LOCS. = 8 WAY LEAVE
477 001704 000000 0 0 ;END OF TABLE TERMINATOR

```



```

478
479 001706          BEGINI
480 001706 012786 001100      MOV  #STACK.SP          ISETUP THE STACK PCINTER
481 001712 012737 012062 000020  MOV  #SSCOPE;#IOTVEC    I IOT VECTOR FOR SCOPE ROUTINE
482 001720 012737 000340 000022  MOV  #340;#IOTVEC*2     I LEVEL 7
483 001726 000067 177350      CLR  $STSNM             I INITIALIZE THE TEST NUMBER
484 001732 012737 012634 000030  MOV  #SHLT;#EMTVEC      IEMT VECTOR FOR HLT(ERROR) ROUTINE
485 001740 012737 000340 000032  MOV  #340;#EMTVEC*2     I LEVEL 7
486 001746 012737 013556 000034  MOV  #STRAP;#TRAPVEC    I TRAP VECTOR FOR TRAP CALLS
487 001754 012737 000340 000036  MOV  #340;#TRAPVEC*2    I LEVEL 7
488 001762 012737 013606 000024  MOV  #SPWRD;#PWRVEC     I POWER FAILURE VECTOR
489 001770 012737 000340 000026  MOV  #340;#PWRVEC*2     I LEVEL 7
490 001776 000067 177276      CLR  SPASS             I CLEAR THE PASS COUNT
491 002002 000067 177276      CLR  SICNT             I INITIALIZE THE ITERATION COUNTER
492 002006 000067 010276      CLR  $TIMES           I INITIALIZE NUMBER OF ITERATIONS
493 002012 100067 177276      CLR  SERFLG           I CLEAR THE ERROR FLAG
494 002016 000067 177270      CLR  SERTL            I CLEAR THE ERROR COUNT
495 002022 000067 010746      CLR  SESCAPE          I CLEAR THE ESCAPE CN ERROR ADDRESS
496
497 002026 000037 001630      CLR  #USERTYPE        I SET USER SELECTION INDICATOR
498                               I TO ZERO INDICATING PROGRAM
499                               I TABULATION
500 002032 000037 002304      CLR  #SKT11           I CLEAR KT11 PRESENCE FLAG
501 002036 000037 001420      CLR  #SSET0           I CLEAR THE OFFSET
502 002042 000037 001422      CLR  #SSET1           I TABLE LOCATIONS FCR
503 002046 000037 001424      CLR  #SSET2           I THE KT11 OPTION
504 002052 000037 001426      CLR  #SSET3
505 002056 000037 001430      CLR  #SSET4
506 002062 000037 001432      CLR  #SSET5
507 002066 000037 001434      CLR  #SSET6
508 002072 000037 001436      CLR  #SSET7
509 002076 000037 001440      CLR  #SSET10
510 002102 000037 001442      CLR  #SSET11
511 002106 000037 001444      CLR  #SSET12
512 002112 000037 001450      CLR  #NTER0           I CLEAR THE INTERLEAVE TABLE
513 002116 000037 001452      CLR  #NTER1           I ENTRY LOCATIONS
514 002122 000037 001454      CLR  #NTER2
515 002126 000037 001456      CLR  #NTER3
516 002132 000037 001460      CLR  #NTER4
517 002136 000037 001462      CLR  #NTER5
518 002142 000037 001464      CLR  #NTER6
519 002146 000037 001466      CLR  #NTER7
520 002152 000037 001470      CLR  #NTER10
521 002156 000037 001472      CLR  #NTER11
522 002162 000037 001474      CLR  #NTER12
523 002166 000037 001636      CLR  #LEAFCNT        I CLEAR NO. OF ABORTS PER NO. OF
524                               I CONSECUTIVE LOCs, TESTED FLAG
525 002172 000037 001642      CLR  #CRU40           I CLEAR PROCESSOR INDICATOR FLAG
526 002176 013746 000004      MOV  #4,-(SP)         I SAVE CONTENTS OF LOC. 4
527 002202 013746 000010      MOV  #10,-(SP)        I SAVE CONTENTS OF LOC. 10
528 002206 012737 002226 000010  MOV  #13;#RESVEC      I SET UP FOR 'SPL' TRAP ADDRESS
529 002214 012737 000340 000012  MOV  #340;#RESVEC*2   I SET UP FOR 'SPL' TRAP PS
530 002222 000237              SPL  7                 I ATTEMPT TO SET A PRIORITY LEVEL
531 002224 000403              BR   25               I BRANCH INDICATING WE ARE ON AN

```

```

532
533 002226 022626          1S:  CMP  (SP)+,(SP)+      I11/45 PROCESSOR
534 002230 005237 001642      INC  #CRU40           I RESET THE STACK FROM TRAP
535                               I SET FLAG INDICATING WE ARE ON
536 002234 012637 000010      2S:  MOV  (SP)+,#10      IAN 11/40 PROCESSOR
537 002240 000037 000012      CLR  #N12            I RESTORE CONTENTS CF LOC. 10
538 002244 012737 002324 000004  MOV  #KT1TIMEOUT;#ERRVEC I RESTORE TRAPCATCHER LOC. 12
539 002252 012737 000340 000006  MOV  #340;#ERRVEC*2    I SET UP KT TIMEOUT ADDRESS
540 002260 005777 175720      TST  #SR0             I SET UP KT TIMEOUT PS
541 002264 000077 175714      CLR  #SR0             I KT11 ARE YOU THERE?
542                               I YES - INITIALIZE IT IN CASE
543 002270 013700 175700      MOV  #SWR,R0         I USER DOESN'T WANT IT
544 002274 006300          R0      I GET SWR CONTENTS
545 002276 103700          ASL  R0              I MOVE BIT06 TO BIT27 POSITION
546                               I (KT11 PRESENT (OBVIOUSLY) IF
547                               I WE REACH THIS INSTRUCTION
548                               I DOES USER WANT IT?
549 002302 005327          DEC  (PC)+           I BRANCH IF NO
550 002304 000000          0      I YES = SET KT11 FLAG
551                               I CONTAINS A -1 IF KT11 OPTION
552 002306 004737 013124      5KT11: JSR  PC;#SSIZE      I IS PRESENT
553 002312 000077 175712      CLR  #KPAR0         I SEE HOW MUCH MEMORY IS AVAILABLE
554 002316 005277 175662      INC  #SR0            I CLEAR PAGE 0 OFFSET REGISTER
555 002322 000401          BR   GO              I TURN ON MEMORY MANAGEMENT
556 002324 022626          KTIMEOUT: CMP  (SP)+,(SP)+  I SKIP NEXT INSTRUCTION
557                               I RESET THE STACK FROM TIMEOUT
558                               I (KT11 NOT PRESENT, THEREFORE
559                               I ONLY 80 BELOW 20K
559 002326 012637 000004      GOI  MOV  (SP)+,#4     I RESTORE CONTENTS CF LOC. 4
560 002332 000037 000006      CLR  #6              I RESTORE CONTENTS CF LOC. 6
561 002336 004337 011506      JSR  R3;#INITIALIZE I SET UP TO BEGIN TESTING
562 002342 016703 177046      MOV  $TPAD,R3        I SET UP FOR MEMORY TABLE CREATION
563 002346 016702 176764      MOV  $PEGAD,R2       I SET UP FOR PARITY TABLE CREATION
564 002352 016700 177040      MOV  $SETAD,R0        I SET UP FOR OFFSET TABLE CREATION
565                               I THIS TABLE ONLY HAS EFFECT IF
566                               I MEMORY MGMT IS TURNED ON
567 002356 016705 177064      MOV  NTERAD,R5       I SET UP FOR INTERLEAVE TABLE
568                               I CREATION (8 - WAY INTERLEAVE
569                               I CAPABILITY EXISTS)
570
571 ;
572 ; *****
573 ;
574 ; LET'S DETERMINE IF SEVERAL REGISTERS EXIST, FOR EXAMPLE,
575 ;
576 ;     172100 GOVERNING CORE MEMORY    0 - 8K
577 ;     & 172102 GOVERNING MOS MEMORY   8 - 16K
578 ;     & 172112 GOVERNING CORE MEMORY 40 - 56K
579 ;
580 ; IF WE WANT TO PRESELECT ONE OF THEM OR CREATE A TABLE OF ALL THOSE
581 ; AVAILABLE AND CARRY ON TESTING FROM THE TABLE
582 ;
583 ; NOTE: SEE DOCUMENT CONCERNING TABLE APPEARANCES AS A
584 ; FUNCTION OF MEMORY MANAGEMENT (KT11 OPTION) BEING
585 ; ENABLED OR DISABLED DURING PROGRAM EXECUTION
586 ;
587 ; *****

```

```

586 002362 032737 010000 177570      BIT      #BIT12,0#SWR      IDDER THE USER WISH TO SELECT THE
587                                     BIT      FINDONE          IREGISTER?
588 002370 001445                                     BEQ      FINDONE          IBRANCH IF NO
589 002372 104400      MSGOYPI  TYPE      ,,4          ITYPE ASCII STRING
590 002372 104400      BR        643          IGET OVER THE ASCII
591 002376 030433      I,ASCII  <15><12>"TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN "
592
593 002466 104406 001340      645I    ACCEPT,SREG0     IPICK UP THE DESIRED REGISTER
594 002466 104406 001340      ACCEPY,SREG0     IFROM THE TELETYPE AND STORE
595                                     INC      #USERTYPE      IIN FIRST TABLE LOCATION
596 002472 013746 000004      MOV      #4,-(SP)      ISET FLAG INDICATING USER SELECTION
597 002472 013746 000004      BR       NEXT1         ISAVE CONTENTS OF LOC. 4
598 002502 000404      FINDONEI  MOV      #172100,(R2) ISKIPE THE NEXT INSTRUCTIONS
599 002504 012712 172100      BR       NEXT1         IMOVE 1ST POSSIBLE PARITY
600                                     MOV      #4,-(SP)      IREGISTER INTO SREG0
601                                     MOV      #INTERTABLE,R4 IPUSH CONTENTS OF LOC.4 ONTO STACK
602 002510 013746 000004      NEXT1I  MOV      #INTERTABLE,R4 IINITIALIZE INTERLEAVE TABLE
603 002514 012704 001644      TST     #SKT11        IPOINTER
604                                     BEQ     #5            IKT11 ARE YOU THERE?
605 002520 005737 002304      CLR     #KPAR1        IBRANCH IF NO
606 002524 001402 000000      TST     #SKT11        IRESET PAGE 1 ADDRESS REGISTER
607 002526 005077 179500      BEQ     #5            IBEFORE TESTING NEXT PARITY
608                                     CLR     #KPAR1        ICONTROL REGISTER
609                                     MOV     #NOREG,0#4    ISET PARITY TIMEOUT VECTOR SERVICE ADDRESS
610 002532 012737 003144 000004 56I  MOV     #172130,(R2) IIS THE ADDRESS IN BOUNDS?
611 002540 022712 172130      CMP     #129          IBRANCH IF YES
612 002544 100002 129      BPL     #NOMORE       IOTHERWISE - TERMINATE TABLE
613 002546 000137 003172      JMP     #R2           IYES - IS THIS REGISTER PRESENT?
614 002552 005772 000000 125I  TST     R7,0#PARTST  ICHECK OUT FOR FATAL ERRORS
615 002556 004737 003222      JSR     R7,0#PARTST
616
617 ;
618 ;
619 ;
620 ;
621 ;
622 002562 012737 003112 000004      MOV     #PARCORE,0#4  ISET MEMORY TIMEOUT VECTOR
623                                     MOV     #13700,0#MEMAD ISERVICE ADDRESS
624 002570 012737 013700 001640      MOV     (R4),0#BLKCNT ISET UP A STARTING ADDRESS
625 002576 011437 001632      MOV     #R4,0#BLKCNT ISET A COUNTER FOR CONSECUTIVE
626                                     MOV     #R4,0#BLKCNT ILOCATION CHECKS TO COVER MEMORY
627                                     TST     #SKT11        INTERLEAVING
628 002602 005737 002304      BPL     #15          ISHOULD I LOOK ABOVE 20K?
629 002606 100013      ADD     #10000,0#MEMAD IBRANCH IF NO
630 002610 062737 010000 001640      MOV     #MEMAD,0#RESTOREBASE ISTEP UP TO A PAGE 1 BASE ADDRESS
631                                     ADD     #140,(R0)      IF MEMORY MANAGEMENT TURNED ON
632 002616 013737 001640 001634      MOV     (R0),0#KPAR1 ISAVE PAGE 1 BASE ADDRESS
633 002624 062710 000140 25I  ADD     (R0),0#KPAR1 ISET UP AN OFFSET FOR KPAR1
634 002630 011077 179376      BR      #4000,0#MEMAD ISET OFFSET IN PAGE 1 REGISTER
635 002634 000406      BR      #9          ISKIPE NEXT 2 INSTRUCTIONS
636 002636 062737 004000 001640 15I  ADD     #4000,0#MEMAD ISTEP UP TO NEXT BANK
637 002644 013737 001640 001634      MOV     #MEMAD,0#RESTOREBASE ISAVE INITIAL MEMORY ADDRESS
638 002652 005777 176762 95I  TST     #MEMAD        IIS THIS MEMORY AVAILABLE?
639 002656 013713 001640      MOV     #MEMAD,(R3)   IYES - STORE THIS MEMORY LOCATION

```

```

640 002662 004737 003430      JSR     R7,0#ABORT    INOW LET'S SEE IF IT'S PARITY
641                                     MOV     #MEMAD,0#MEMAD IMEMORY CORRESPONDING TO THE
642                                     TST     #SKT11        IPARITY REGISTER HAVE FOUND
643 002666 005737 002304      BPL     #49          IKT11 ARE YOU THERE?
644 002672 100034      DEC     #BLKCNT       IBRANCH IF NO
645 002674 005337 001632      DEC     #BLKCNT       IDECREASE CONSECUTIVE
646                                     TST     #BLKCNT       ILOCATION COUNTER
647 002700 005737 001632      TST     #BLKCNT       IARE WE DONE CHECKING
648                                     BEQ     #6           ICONSECUTIVE LOCATIONS?
649 002724 001404      ADD     #2,0#MEMAD   IBRANCH IF YES
650 002706 062737 000002 001640      BR      #9          ISTEP UP 1 LOCATION
651 002714 000756      BR      #9          IGO BACK TO TEST WITH THIS
652                                     MOV     #RESTOREBASE,0#MEMAD ILOCATION
653 002716 013737 001634 001640 65I  MOV     #RESTOREBASE,0#MEMAD IRESTORE PAGE 1 BASE ADDRESS
654                                     ADD     #4,R4         IBEFORE GOING BACK TO INCREASE
655                                     ADD     #4,R4         IOFFSET
656 002724 062704 000004      ADD     #4,R4         ISTEP TABLE POINTER UP FOR
657                                     TST     (R4)         INEXT VALUE OF CONSECUTIVE
658 002730 005714      BEQ     #10          ILOCATION TO BE CHECKED
659 002732 031405      BEQ     #10          IARE THERE ANY MORE?
660 002734 011437 001632      MOV     (R4),0#BLKCNT IBRANCH IF NO
661                                     CLR     #LEAFCNT     ISTORE THIS VALUE CF CONSECUTIVE
662 002740 005037 001636      CLR     #LEAFCNT     ILOCATION CHECKS
663 002744 000742      BR      #9          ICLEAR INTERLEAVE VALUE HOLDER
664                                     BR      #9          IBEFORE RETESTING
665 002746 012704 001644 105I  MOV     #INTERTABLE,R4 IGO BACK TO TEST WITH THIS VALUE
666                                     MOV     (R4),0#BLKCNT IOF CONSECUTIVE LOCATIONS
667 002752 011437 001632      MOV     (R4),0#BLKCNT IINITIALIZE INTERLEAVE TABLE
668                                     CLR     #LEAFCNT     IPOINTER
669 002756 005037 001636      CLR     #LEAFCNT     IRESET THE CONSECUTIVE
670                                     BR      #25          ILOCATION COUNTER
671 002762 000720      BR      #25          ICLEAR INTERLEAVE VALUE HOLDER
672                                     BR      #25          IBEFORE RETESTING
673 002764 022737 157700 001640 45I  CMP     #157700,0#MEMAD IGO BACK TO INCREASE OFFSET
674 002772 001434      BEQ     #85          IAND TEST
675 002774 005337 001632      DEC     #BLKCNT       IARE WE UP TO 20K YET?
676 002776 005337 001632      DEC     #BLKCNT       IBRANCH IF YES
677 002778 005337 001632      TST     #BLKCNT       IDECREASE CONSECUTIVE
678 002780 005337 001632      TST     #BLKCNT       ILOCATION COUNTER
679 002782 005337 001632      TST     #BLKCNT       IARE WE DONE CHECKING CONSEC-
680                                     BEQ     #75          IUTIVE LOCATIONS?
681 002784 005337 001632      BEQ     #75          IBRANCH IF YES
682 002786 005337 001632      ADD     #2,0#MEMAD   ISTEP UP 1 LOCATION
683 002788 005337 001632      BR      #9          IGO BACK TO TEST WITH THIS
684                                     MOV     #RESTOREBASE,0#MEMAD ILOCATION
685 002790 013737 001634 001640 75I  MOV     #RESTOREBASE,0#MEMAD IRESTORE INITIAL MEMORY
686                                     ADD     #4,R4         IADDRESS BEFORE GOING BACK TO
687 002792 062704 000004      ADD     #4,R4         ISTEP UP TO NEXT BANK
688 002794 062704 000004      ADD     #4,R4         ISTEP TABLE POINTER UP FOR
689                                     TST     (R4)         INEXT VALUE OF CONSECUTIVE
690 002796 005714      BEQ     #115         ILOCATIONS TO BE CHECKED
691 002798 005714      BEQ     #115         IARE THERE ANY MORE?
692 002800 005714      BEQ     #115         IBRANCH IF NO
693 002802 011437 001632      MOV     (R4),0#BLKCNT ISTORE THIS VALUE CF

```

```

694                                     ;CONSECUTIVE LOCATION CHECKS
695 003040 005037 001636          CLR  #BLEAFCNT          ;CLEAR INTERLEAVE VALUE HOLDER
696                                     ;BEFORE RETESTING
697 003044 000702          BR    9S                ;GO BACK TO STEP UP TO THE
698                                     ;NEXT BANK TO CONDUCT TESTING
699 003046 012704 001644          11SI MOV  #INTERTABLE,R4      ;INITIALIZE INTERLEAVE
700                                     ;TABLE POINTER
701 003052 011437 001632          MOV  (R4),#BLKCNT        ;RESET THE CONSECUTIVE
702                                     ;LOCATION COUNTER
703 003056 005037 001636          CLR  #BLEAFCNT          ;CLEAR INTERLEAVE VALUE HOLDER
704                                     ;BEFORE RETESTING
705 003062 000665          BR    1S                ;GO BACK TO STEP UP TO NEXT
706                                     ;BANK TO CONDUCT TESTING
707 003064 011237 001622          8SI  MOV  (R2),#PARITY      ;STORE THE BAD REGISTER WITH
708                                     ;NO PARITY MEMORY
709 003070 104004          HLT  +4                ;NO PARITY MEMORY FOUND
710                                     ;BELOW 20K!!!!!!!
711 003072 005737 001630          TST  #USERTYPE          ;DID USER SELECT REGISTER?
712 003076 001402          BEQ  5S                ;BRANCH IF NO
713 003100 000137 002372          JMP  #MSGTYP           ;GO BACK TO RETYPE MESSAGE FOR
714                                     ;USER RESPONSE
715 003104 062712 000002          3SI  ADD  #2,(R2)        ;PLACE NEXT POSSIBLE REGISTER
716                                     ;INTO SAME TABLE LOCATION
717 003110 000601          BR    NEXT1            ;GO BACK TO TEST THIS REGISTER
718 003112 022626          PARCOR: CMP  (SP),#(SP)+ ;RESET STACK FROM MEMORY TIMEOUT
719 003114 011237 001622          MOV  (R2),#PARITY      ;STORE THE REGISTER THAT EN-
720                                     ;COUNTERED A POSSIBLE HOLE IN
721                                     ;MEMORY
722 003120 104007          HLT  +7                ;A POSSIBLE HOLE IN MEMORY EXISTS
723                                     ;WITH NO PARITY BELOW IT!!!!!!!
724 003122 005737 001630          TST  #USERTYPE          ;DID USER SELECT REGISTER?
725 003126 001402          BEQ  4S                ;BRANCH IF NO
726 003130 000137 002372          JMP  #MSGTYP           ;GO BACK TO RETYPE MESSAGE FOR
727                                     ;USER RESPONSE
728 003134 062712 000002          4SI  ADD  #2,(R2)        ;PLACE NEXT POSSIBLE REGISTER
729                                     ;INTO SAME TABLE LOCATION
730 003140 000137 002514          NOREG: JMP  #NEXT1            ;GO BACK TO TEST THIS REGISTER
731 003144 022626          CMP  (SP),#(SP)+      ;RESET STACK FROM REGISTER TIMEOUT
732 003146 005737 001630          TST  #USERTYPE          ;DID THE USER SELECT THE REGISTER?
733 003152 001403          BEQ  1S                ;BRANCH IF NO
734 003154 104006          HLT  +6                ;YES - USER SELECTED REGISTER NOT
735                                     ;PRESENT ON SYSTEM
736 003156 000137 002372          JMP  #MSGTYP           ;GO BACK TO RETYPE MESSAGE
737 003162 062712 000002          1SI  ADD  #2,(R2)        ;STEP UP TO NEXT PARITY REGISTER
738                                     ;AT SAME TABLE LOCATION
739 003166 000137 002514          JMP  #NEXT1            ;PREVIOUS PARITY REGISTER NOT
740                                     ;PRESENT - SEE IF THE NEXT ONE IS
741 003172 012637 000004          NOMORE: MOV  (SP),#4    ;RESTORE CONTENTS OF LOC. 4
742 003176 000012          CLR  (R2)              ;CALL DONE TABLE CREATION
743                                     ;END IT WITH A '0'
744 003200 000137 003620          JMP  #START            ;START RUNNING PROGRAM WITH
745                                     ;TABLE CONTENTS
746 ;*****
747 ;

```

```

748                                     ;THE FOLLOWING ROUTINE WILL CREATE A 2 LOCATION MEMORY MAP AT
749                                     ;THE HIGH END OF A 4K CORE SECTION. THIS 2 LOCATION MAP WILL
750                                     ;INITIALLY BE USED TO DETERMINE WHERE/IF PARITY MEMORY
751                                     ;PRESIDES AND LATER FOR SUBSEQUENT PROGRAM TESTING OF A REGISTER
752                                     ;
753 ;*****
754 003204 162700 000376          COMPUTI: SUB  #370,R0      ;DROP DOWN SO AS NOT TO
755                                     ;DESTROY ABS LOADER
756 003210 010001          MOV  R0,R1            ;R1 CONTAINS BEGINNING ADDRESS
757                                     ;OF MEMORY MAP
758 003212 020020          CMP  R0,(R0)          ;STEP 00 TO NEXT ADDRESS
759 003214 010011          MOV  R0,R1            ;1ST MEMORY LOCATION
760 003216 011110          MOV  R0,R0            ;2ND MEMORY LOCATION
761 003220 000203          RTS  R3              ;RETURN TO TEST A CAPI
762                                     ;WITH CONTENTS OF THESE 2 LOCS.
763 ;*****
764 ;
765 ;
766 ;THIS ROUTINE WILL CHECK IF THE PARITY REGISTER IS STATICALLY IN
767 ;GOOD OPERATION FOR TESTING TO BE CONDUCTED
768 ;
769 ;*****
770 003222 011267 176374          PARTSY: MOV  (R2),PARITY ;GET PARITY REGISTER TO BE USED
771                                     ;
772 ;TEST 1 SET BIT0 (USED) OF PARITY REGISTER
773 ;*****
774 003226 000004          TST1: SCOP
775 003230 052777 000001 176384          BIS  #BIT0,PARITY      ;DID IT SET?
776 003236 032777 000001 176396          BIT  #BIT0,PARITY      ;YES
777 003244 001001          BNE  +4                ;NO - FATAL ERROR TO PROGRAM!
778 003246 104002          HLT  +2                ;
779 ;*****
780 ;TEST 2 CLEAR BIT0 (USED) OF PARITY REGISTER
781 ;*****
782 003250 000004          TST2: SCOP
783 003252 042777 000001 176342          BIC  #BIT0,PARITY      ;DID IT CLEAR?
784 003260 032777 000001 176334          BIT  #BIT0,PARITY      ;YES
785 003266 001401          BEQ  +4                ;NO - FATAL ERROR TO PROGRAM!
786 003270 104002          HLT  +2                ;
787 ;*****
788 ;TEST 3 SET AND CLEAR BIT2 (USED) OF PARITY REGISTER
789 ;*****
790 003272 000004          TST3: SCOP
791 003274 052777 000004 176320          BIS  #BIT2,PARITY      ;DID IT SET?
792 003302 032777 000004 176312          BIT  #BIT2,PARITY      ;YES
793 003310 001001          BNE  +4                ;NO - FATAL ERROR TO PROGRAM!
794 003312 104002          HLT  +2                ;
795 003314 042777 000004 176300          BIC  #BIT2,PARITY      ;DID IT CLEAR?
796 003322 032777 000004 176272          BIT  #BIT2,PARITY      ;YES
797 003330 001401          BEQ  +4                ;NO - FATAL ERROR TO PROGRAM!
798 003332 104002          HLT  +2                ;
799 ;*****
800 ;TEST 4 TEST RESET ON BITS 0, 2 AND 15
801 ;*****

```

```

802 003334 000004          TST4: SCOPE
803 003336 005737          TST  WHICH1  ##SKT11          JKT11 ON?
804 003342 100415          BHI          WHICH1          IBRANCH IF YES AND DON'T DO
805                                     ITHIS TEST BECAUSE THE 'RESET'
806                                     IWILL GLOBBER SEGMENTATION
807 003344 052777 100005 174250      BIS          #100005,PARITY
808 003352 000005          RESET        IEXPECT BITS 0, 2 AND 19 TO CLEAR
809 003354 032777 100005 174240      BIT          #100005,PARITY      DID THEY CLEAR?
810 003362 001404          BEQ          #12                IYES
811 003364 042777 100005 174230      BIC          #100005,PARITY      JNO - CLEAR OUT REGISTER AS A
812                                     IPRECAUTION
813 003372 104005          HLT          #5                IRESET DOESN'T WORK
814                                     ;*****
815                                     ;TEST 5 WHICH OPTION IS ABOUT TO BE TESTED
816                                     ;*****
817 003374 000004          TST5: SCOPE
818                                     ;*****
819 003376 052777 007740 174216      WHICH1: BIS  #7740,PARITY        IIS AN OLD MS11 OPTION
820                                     IWITH NO ADDRESS BITS
821                                     IABOUT TO BE TESTED?
822 003404 032777 007740 174210      BIT          #7740,PARITY      IADDRESS BITS ABLE TO BE SET?
823 003412 001402          BEQ          15                IBRANCH IF NO INDICATING MS11
824 003414 000004          SCOPE
825 003416 000207          RTS          R7                IRETURN TO NORMAL FLOW
826 003420 005237 001626          IS: INC      ##MSREGFLAG       ISET FLAG INDICATING MS11 OPTION
827                                     IWITH NO ADDRESS BITS
828 003424 000004          SCOPE
829 003426 000207          RTS          R7                IRETURN TO NORMAL FLOW
830
831                                     ;*****
832                                     ;
833                                     ;THE FOLLOWING ROUTINE WILL TAKE EACH 1K BANK OF MEMORY
834                                     ;THAT IS AVAILABLE AND PERFORM A DATI IN IT
835                                     ;TO DETERMINE IF PARITY EXISTS THERE, THIS ROUTINE IS
836                                     ;ONLY USED DURING TABLE CREATION
837                                     ;*****
838                                     ;
839 003430 010546          ABORT: MOV   R5,-(SP)          ISAVE R5 CONTENTS ON STACK
840 003432 010046          MOV   R0,-(SP)          ISAVE R0 CONTENTS ON STACK
841 003434 011300          MOV   (R3),R0           IGET THE MEMORY LOCATION
842                                     IJUST DETERMINED
843 003436 004337 003204          JSR   R3,##COMPUT        ICOMPUTE AN AREA IN THIS BANK
844                                     IFOR DETERMINING PARITY MEMORY
845 003442 011267 176154          MOV   (R2),PARITY        IGET THE PARITY REGISTER JUST
846                                     IFOUND AND TEST WITH IT
847                                     ;*****
848                                     ;TEST A DATI IN THIS BANK
849                                     ;*****
850                                     ;11/45 *** ROM STATE 221 ***
851                                     ;
852                                     ;
853 003446 012705 011450          MOV   ##VECT,R5          ISET UP SERVICE ROUTINE ADDRESS
854 003452 004015          JSR   R0,(R5)           ISET UP PARITY VECTOR SERVICE
855 003454 003500          ONETRY          IROUTINE ADDRESS
    
```

```

856 003456 011100          MOV   @R1,R0           ISET UP FOR A DATO
857 003460 010010          MOV   R0,R0           IDO THE DATO
858 003462 010030          MOV   R0,@(R0)        IDO A DATI
859 003464 042777 000005 176130      BIC          #BIT2|BIT0,PARITY  IWRITE NORMAL AND C|SABLE
860 003472 012600          MOV   (SP)+,R0        IRESTORE R0 CONTENTS
861 003474 012605          MOV   (SP)+,R5        IRESTORE R5 CONTENTS
862 003476 000207          RTS          R7         INOT PARITY MEMORY!
863                                     IRETURN TO TEST AT NEXT
864                                     IINCREMENT
865 003500 042777 000005 176114      ONETRY: BIC  #BIT2|BIT0,PARITY  IWE HAVE PARITY MEMORY - PROCEED
866 003506 016600 000004          MOV   4(SP),R0        IRESTORE R0 CONTENTS
867 003512 016605 000006          MOV   6(SP),R5        IRESTORE R5 CONTENTS
868 003516 005237 001636          INC   ##LEAFCNT        IINCREMENT INTERLEAVE COUNTER
869 003522 022737 000003 001636      CMP   #3,##LEAFCNT     I3 ABORTS REACHED?
870 003530 001403          BEQ   15                IBRANCH IF YES
871 003532 062706 000010          ADD   #10,SP          IBYPASS JUNK ON STACK
872 003536 000207          RTS          R7         IRETURN TO TEST AT NEXT INCREMENT
873 003540 022626          IS: CMP   (SP)+,(SP)+  IPOP STACK BACK FROM PARITY ABORT
874 003542 012600          MOV   (SP)+,R0        IRESTORE R0 CONTENTS
875 003544 012605          MOV   (SP)+,R5        IRESTORE R5 CONTENTS
876 003546 005726          TST   (SP)+           IPOP STACK ONCE FOR ABORT ROUTINE
877                                     IENTRY
878 003550 005737 001630          TST   ##USERTYPE       IDID USER TYPE IN REGISTER?
879 003554 001404          BEQ   23                IBRANCH IF NO
880 003556 016415 000002          MOV   2(R4),(R5)       ISET INTERLEAVE VALUE INTO
881                                     ITABLE
882 003562 000137 003620          JMP   ##START          IAND LOCK ON THE USER SELECTED
883                                     IREGISTER FOR TESTING
884 003566 005723          2S: TST   (R3)+        IUSER DIDN'T SELECT - SO STEP UP
885                                     ITO NEXT MEMORY TABLE LOCATION
886 003570 005720          TST   (R0)+           ISTEP UP TO NEXT OFFSET TABLE
887                                     ILOCATION - THIS TABLE WILL ONLY
888                                     IBE APPLICABLE IF MEMORY MGMT
889                                     IIS TURNED ON
890 003572 012212          MOV   (R2)+,(R2)       ISET NEXT POSSIBLE REGISTER INTO
891 003574 042712 000002          ADD   #2,(R2)          INEXT REGISTER TABLE LOCATION
892 003600 016425 000002          MOV   2(R4),(R5)+     ISET INTERLEAVE VALUE INTO
893                                     ITABLE
894 003604 005037 001636          CLR   ##LEAFCNT        IRESET NO. OF ABORTS COUNTER
895 003610 005037 001626          CLR   ##MSREGFLAG     ICLEAR PARITY TYPE INDICATOR
896 003614 000137 002514          JMP   ##NEXT1         IGO BACK TO CHECK NEXT POSSIBLE
897                                     IPARITY REGISTER
898
899                                     ;*****
900                                     ;
901                                     ;
902                                     ;IF WE HAVE REACHED THIS POINT IN THE PROGRAM THEN,
903                                     ;
904                                     ;
905                                     ;
906                                     ;
907                                     ;
908                                     ;
909                                     ;
910                                     ;
911                                     ;
912                                     ;
913                                     ;
914                                     ;
915                                     ;
916                                     ;
917                                     ;
918                                     ;
919                                     ;
920                                     ;
921                                     ;
922                                     ;
923                                     ;
924                                     ;
925                                     ;
926                                     ;
927                                     ;
928                                     ;
929                                     ;
930                                     ;
931                                     ;
932                                     ;
933                                     ;
934                                     ;
935                                     ;
936                                     ;
937                                     ;
938                                     ;
939                                     ;
940                                     ;
941                                     ;
942                                     ;
943                                     ;
944                                     ;
945                                     ;
946                                     ;
947                                     ;
948                                     ;
949                                     ;
950                                     ;
951                                     ;
952                                     ;
953                                     ;
954                                     ;
955                                     ;
956                                     ;
957                                     ;
958                                     ;
959                                     ;
960                                     ;
961                                     ;
962                                     ;
963                                     ;
964                                     ;
965                                     ;
966                                     ;
967                                     ;
968                                     ;
969                                     ;
970                                     ;
971                                     ;
972                                     ;
973                                     ;
974                                     ;
975                                     ;
976                                     ;
977                                     ;
978                                     ;
979                                     ;
980                                     ;
981                                     ;
982                                     ;
983                                     ;
984                                     ;
985                                     ;
986                                     ;
987                                     ;
988                                     ;
989                                     ;
990                                     ;
991                                     ;
992                                     ;
993                                     ;
994                                     ;
995                                     ;
996                                     ;
997                                     ;
998                                     ;
999                                     ;
1000                                    ;
    
```

```

910 ;
911 ; LOCATION 1342 UP TO 1364 WILL CONTAIN ANY MORE
912 ; REGISTERS FOUND
913 ;STMP@ (LOCATION 1366) WILL CONTAIN AN ADDRESS IN THE BANK
914 ; THAT HAS PARITY ASSOCIATED WITH THE PARITY
915 ; REGISTER IN SREG@
916 ;
917 ; LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING
918 ; MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS
919 ;SSET@ (LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED
920 ; WITH THE CORRESPONDING VALUE IN STMP@
921 ;
922 ; LOCATION 1422 TO 1444 WILL CONTAIN THE CORRESPONDING
923 ; OFFSET VALUES FOR THE OTHER REGISTERS
924 ;
925 ;INTER@ (LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED
926 ; WITH THE PARITY REGISTER IN SREG@
927 ;
928 ; LOCATION 1452 TO 1474 WILL CONTAIN THE CORRESPONDING
929 ; INTERLEAVE FACTORS FOR THE OTHER REGISTERS
930 ;
931 ;NOTE: TO UNDERSTAND THE TABLE SETUP SEE THE DOCUMENT
932 ;
933 ;
934 ;*****
935 ;*****
936 ;*****
937 ;*****
938 ;*****
939 ;*****
940 ;*****
941 ;
942 ;THE REST OF THE PROGRAM WILL TEST ALL PARITY ABORTS
943 ;USING EITHER THE PARITY REGISTER TYPED BY THE USER OR THOSE
944 ;ENCOUNTERED IN THE TABLE PREVIOUSLY GENERATED.
945 ;
946 ;*****
947 003620 005777 175512 STARTI TST @SREGAD ;ARE WE AT END OF TABLE?
948 003624 001401 BEQ 15 ;BRANCH IF YES
949 003626 000402 BR 25 ;OTHERWISE SKIP NEXT INSTRUCTION
950 003630 000137 011464 1SI JMP @SRESTART ;SET UP TO START TABLE OVER!!
951 003634 017767 175476 2SI MOV @SREGAD,PARITY ;GET PARITY REGISTER TO BE USED
952 003642 162767 000002 175432 SUB #2,$STSNM ;DECREASE TEST NO. FOR SCOPE
953 ;
954 ; LOOPING IN CASE TEST #4 WAS
955 ; DONE BECAUSE TEST #4 WAS
956 ; DONE THROWS $STSNM COUNT OFF
957 ; BY 2
958 003650 005737 002304 TST @#SKT11 ;K11 ARE YOU THERE?
959 003654 100006 BPL 35 ;BRANCH IF NO
960 003664 062767 000001 175410 ADD #1,$STSNM ;GET THE OFFSET NEEDED
961 ; SET TEST NO. TO A CORRECT
962 ; VALUE! TEST #4 WAS NOT
963 003672 017700 175516 3SI MOV @STMPAD,R0 ;EXECUTED
; GET PARITY MEMORY ASSOCIATED

```

```

964 ;
965 003676 004337 003204 JSR R3,@#COMPUT ;WITH THE PARITY REGISTER
966 ; COMPUTE AN AREA IN THIS BANK
967 003702 010137 001476 MOV R1,@#NEWSTK ;FOR TESTING
968 ; SET UP A NEW STACK POINTER
969 ; FOR STACK OPERATIONS IN CASE
970 ; NO PARITY MEMORY RESIDES IN
971 ; LOWER 4K
972 003706 004737 003376 JSR PC,@#WHICH1 ;DETERMINE IF WE ARE ABOUT TO
973 ; TEST AN OLD MOS DESIGN!
974 003712 012705 011450 MOV #VEGSET,R5 ;SET UP SERVICE ROUTINE ADDRESS
975 ;*****
976 ;TEST 6 TEST (ADDRESS) SM@,DM3 MOV INSTRUCTION
977 ;*****
978 ;TST6: SCOPE
979 ;
980 ; 11/45 **** ROM STATE 221 ****
981 ;
982 003720 004015 11/40 **** ROM STATE 207 ****
983 003722 003752 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
984 003724 011100 A ;ROUTINE ADDRESS
985 003726 010010 MOV @R1,R0 ;SET UP FOR DATO
986 003730 012737 003740 001332 MOV R0,@R0 ;DO THE DATO
987 ; STORE THE PC THAT SHOULD
988 ; BE PUSHED ON THE STACK
989 ; IF A PARITY ABORT OCCURS
990 ; DO A BATI
991 003736 010030 MOV R0,@(R0)+ ;WRITE NORMAL FOR EMT CALL
992 003740 042777 000004 175634 BIC #BIT2,@PARITY ;IDIDN'T ABORT
993 003746 104001 HLT +1 ;GO TO NEXT TEST
994 003750 000410 BR ,+2 ;WRITE NORMAL AND DISABLE
995 003752 042777 000005 175642 AI BIC #BIT2|BIT0,@PARITY ;CHECK FOR GOOD ABORT
996 003760 004037 011550 JSR R0,@#CHECKLOC ;ABORTED INCORRECTLY
997 003764 104003 HLT +3 ;RESET THE STACK
998 003766 012706 001100 MOV #STACK,SP
999 ;*****
1000 ;TEST 7 TEST (ADDRESS) SM@,DM5 MOV INSTRUCTION
1001 ;*****
1002 ;TST7: SCOPE
1003 ;
1004 ; 11/45 **** ROM STATE 231 ****
1005 ;
1006 1007 003772 000004 11/40 **** ROM STATE 207 ****
1008 ;
1009 1010 003774 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1011 1011 003776 004026 A ;ROUTINE ADDRESS
1012 1012 004000 011100 MOV @R1,R0 ;SET UP FOR DATO
1013 1013 004022 010020 MOV R0,(R0)+ ;DO THE DATO
1014 1014 004004 012737 004014 001332 MOV #+10,@#SGDDAT ;STORE THE PC THAT SHOULD
1015 ; BE PUSHED ON THE STACK
1016 ; IF A PARITY ABORT OCCURS
1017 ; DO A BATI
1018 1016 004012 010030 MOV R0,@(R0) ;WRITE NORMAL FOR EMT CALL
1019 1017 004014 042777 000004 175600 BIC #BIT2,@PARITY ;IDIDN'T ABORT
1020 1018 004022 104001 HLT +1 ;GO TO NEXT TEST
1021 1019 004024 000410 BR ,+2 ;WRITE NORMAL AND DISABLE
1022 1020 004026 042777 000005 175666 AB1 BIC #BIT2|BIT0,@PARITY ;CHECK FOR GOOD ABORT
1023 1021 004034 004037 011550 JSR R0,@#CHECKLOC ;ABORTED INCORRECTLY
1024 1022 004040 104003 HLT +3 ;RESET THE STACK
1025 1023 004042 012706 001100 MOV #STACK,SP

```

```

1018 ;*****
1019 ;TEST 10 TEST (DATA) SH1,DM6 MOV INSTRUCTION
1020 ;*****
1021 TST10: SCOPE
1022 ;
1023 ; 11/45 *** ROM STATE 27 ***
1024 ;
1025 ; 11/40 *** ROM STATE 206 ***
1026 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1027 A1 ;ROUTINE ADDRESS
1028 MOV #R1,R0 ;SET UP FOR DATO
1029 MOV R0,(R0) ;DO THE DATO
1030 MOV #,*10,#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
1031 ;IF A PARITY ABORT OCCURS
1032 MOV #R0-2(R0) ;DO A BATTI
1033 BIC #BIT2,*PARITY ;WRITE NORMAL FOR EMT CALL
1034 HLT #1 ;IDONT' ABORT
1035 BR ,+22 ;GO TO NEXT TEST
1036 BIC #BIT2|BIT0,*PARITY ;WRITE NORMAL AND DISABLE
1037 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABORT
1038 HLT #3 ;ABORTED INCORRECTLY
1039 MOV #STACK,SP ;RESET THE STACK
1040 ;*****
1041 ;TEST 11 TEST (ADDRESS) SH0,DM7 MOV INSTRUCTION
1042 ;*****
1043 TST11: SCOPE
1044 ;
1045 ; 11/45 *** ROM STATE 231 ***
1046 ;
1047 ; 11/40 *** ROM STATE 207 ***
1048 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1049 A2 ;ROUTINE ADDRESS
1050 MOV #R1,R0 ;SET UP FOR A DATO
1051 MOV R0,(R0) ;DO THE DATO
1052 MOV #,*12,#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
1053 ;IF A PARITY ABORT OCCURS
1054 MOV #R0-2(R0) ;DO A BATTI
1055 BIC #BIT2,*PARITY ;WRITE NORMAL FOR EMT CALL
1056 HLT #1 ;IDONT' ABORT
1057 BR ,+22 ;GO TO NEXT TEST
1058 BIC #BIT2|BIT0,*PARITY ;WRITE NORMAL AND DISABLE
1059 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABORT
1060 HLT #3 ;ABORTED INCORRECTLY
1061 MOV #STACK,SP ;RESET THE STACK
1062 ;*****
1063 ;TEST 12 TEST (DATA) SH0,DM2 CMP INSTRUCTION
1064 ;*****
1065 TST12: SCOPE
1066 ;
1067 ; 11/45 *** ROM STATE 175 ***
1068 ;
1069 ; 11/40 *** ROM STATE 267 ***
1070 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1071 B ;ROUTINE ADDRESS
1072 MOV #R1,R0 ;SET UP FOR DATO

```

```

1072 MOV R0,(R0) ;DO THE DATO
1073 MOV #,*10,#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
1074 ;IF A PARITY ABORT OCCURS
1075 ;IF A PARITY ABORT OCCURS
1076 CMP R0,(R0) ;DO A BATTI, DATI
1077 BIC #BIT2,*PARITY ;WRITE NORMAL FOR EMT CALL
1078 HLT #1 ;IDONT' ABORT
1079 BR ,+22 ;GO TO NEXT TEST
1080 BIC #BIT2|BIT0,*PARITY ;WRITE NORMAL AND DISABLE
1081 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABORT
1082 HLT #3 ;ABORTED INCORRECTLY
1083 MOV #STACK,SP ;RESET THE STACK
1084 ;*****
1085 ;TEST 13 TEST (DATA) SH0,DM4 CMP INSTRUCTION
1086 ;*****
1087 TST13: SCOPE
1088 ;
1089 ; 11/45 *** ROM STATE 177 ***
1090 ;
1091 ; 11/40 *** ROM STATE 267 ***
1092 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1093 B0 ;ROUTINE ADDRESS
1094 MOV #R1,R0 ;SET UP FOR DATO
1095 MOV R0,(R0) ;DO THE DATO
1096 MOV #,*10,#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
1097 ;IF A PARITY ABORT OCCURS
1098 ;IF A PARITY ABORT OCCURS
1099 CMP R0,-(R0) ;DO A BATTI, DATI
1100 BIC #BIT2,*PARITY ;WRITE NORMAL FOR EMT CALL
1101 HLT #1 ;IDONT' ABORT
1102 BR ,+22 ;GO TO NEXT TEST
1103 BIC #BIT2|BIT0,*PARITY ;WRITE NORMAL AND DISABLE
1104 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABORT
1105 HLT #3 ;ABORTED INCORRECTLY
1106 MOV #STACK,SP ;RESET THE STACK
1107 ;*****
1108 ;TEST 14 TEST (DATA) SH0,DM6 CMP INSTRUCTION
1109 ;*****
1110 TST14: SCOPE
1111 ;
1112 ; 11/45 *** ROM STATE 177 ***
1113 ;
1114 ; 11/40 *** ROM STATE 267 ***
1115 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1116 B1 ;ROUTINE ADDRESS
1117 MOV #R1,R0 ;SET UP FOR DATO
1118 MOV R0,(R0) ;DO THE DATO
1119 MOV #,*12,#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
1120 ;IF A PARITY ABORT OCCURS
1121 ;IF A PARITY ABORT OCCURS
1122 CMP R0-2(R0) ;DO A BATTI, DATI
1123 BIC #BIT2,*PARITY ;WRITE NORMAL FOR EMT CALL
1124 HLT #1 ;IDONT' ABORT
1125 BR ,+22 ;GO TO NEXT TEST
1126 BIC #BIT2|BIT0,*PARITY ;WRITE NORMAL AND DISABLE
1127 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABORT

```

```
1126 004402 124003          HLT      +3          IABORTED INCORRECTLY
1127 004404 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 15      TEST (DATA) SM0,DM1 CMP INSTRUCTION
;*****
1131 004410 000004          TST15: SCOPE
;          11/45 **** ROM STATE 175 ****
;
;          11/40 **** ROM STATE 267 ****
1135 004412 004015          JSR      R0,(R5)      ISET UP PARITY VECTOR SERVICE
1136 004414 004444          B2          IROUTINE ADDRESS
1137 004416 011100          MOV      @R1,R0      ISET UP FOR DATO
1138 004420 010010          MOV      R0,@R0      IDO THE DATO
1139 004422 012737 004432 001332    MOV      #.+10,@#SGDDAT ISTORE THE PC THAT SHOULD
;          IBE PUSHED ON THE STACK
;          IF A PARITY ABORT OCCURS
1142 004430 020010          CMP      R0,@R0      IDO A DATIP, DATIP
1143 004432 042777 000004 175162    BIC      #BIT2,@PARITY IWRITE NORMAL FOR EMT CALL
1144 004440 104001          HLT      +1          IDIDN'T ABORT
1145 004442 000410          BR       +22         IGO TO NEXT TEST
1146 004444 042777 000005 175190 B2:    BIC      #BIT2|BIT0,@PARITY IWRITE NORMAL AND DISABLE
1147 004452 004037 011550          JSR      R0,#CHECKKLOC ICHECK FOR GOOD ABCRT
1148 004456 104003          HLT      +3          IABORTED INCORRECTLY
1149 004460 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 16      TEST (DATA) SM0,DM3 CMP INSTRUCTION
;*****
1153 004464 000004          TST16: SCOPE
;          11/45 **** ROM STATE 177 ****
;
;          11/40 **** ROM STATE 267 ****
1157 004466 004015          JSR      R0,(R5)      ISET UP PARITY VECTOR SERVICE
1158 004470 004536          B3          IROUTINE ADDRESS
1159 004472 011100          MOV      @R1,R0      ISET UP FOR DATO
1160 004474 010020          MOV      R0,(R0)+    IDO THE DATO
1161 004476 042777 000004 175116    BIC      #BIT2,@PARITY IWRITE NORMAL
1162 004504 011110          MOV      @R1,@R0     IWRITE ADDRESS NORMAL (DATI)
1163 004506 052777 000004 175106    BIS      #BIT2,@PARITY IWRITE OTHER PARITY
1164 004514 012737 004524 001332    MOV      #.+10,@#SGDDAT ISTORE THE PC THAT SHOULD
;          IBE PUSHED ON THE STACK
;          IF A PARITY ABORT OCCURS
1167 004522 020030          CMP      R0,@(R0)+   IDO A DATI, DATIP
1168 004524 042777 000004 175070    BIC      #BIT2,@PARITY IWRITE NORMAL FOR EMT CALL
1169 004532 104001          HLT      +1          IDIDN'T ABORT
1170 004534 000410          BR       +22         IGO TO NEXT TEST
1171 004536 042777 000005 175056 B3:    BIC      #BIT2|BIT0,@PARITY IWRITE NORMAL AND DISABLE
1172 004544 004037 011550          JSR      R0,#CHECKKLOC ICHECK FOR GOOD ABCRT
1173 004550 104003          HLT      +3          IABORTED INCORRECTLY
1174 004552 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 17      TEST (ADDRESS) SM0,DM3 CMP INSTRUCTION
;*****
1178 004556 000004          TST17: SCOPE
;          11/45 **** ROM STATE 221 ****
```

```
1180
1181
;          11/40 **** ROM STATE 264 ****
1182 004560 004015          JSR      R0,(R5)      ISET UP PARITY VECTOR SERVICE
1183 004562 004612          B4          IROUTINE ADDRESS
1184 004564 011100          MOV      @R1,R0      ISET UP FOR A DATO
1185 004566 010010          MOV      R0,@R0      IDO THE DATO
1186 004570 012737 004600 001332    MOV      #.+10,@#SGDDAT ISTORE THE PC THAT SHOULD
;          IBE PUSHED ON THE STACK
;          IF A PARITY ABORT OCCURS
1189 004576 020030          CMP      R0,@(R0)+   IDO A DATI
1190 004600 042777 000004 175014    BIC      #BIT2,@PARITY IWRITE NORMAL FOR EMT CALL
1191 004606 104001          HLT      +1          IDIDN'T ABORT
1192 004610 000410          BR       +22         IGO TO NEXT TEST
1193 004612 042777 000005 175002 B4:    BIC      #BIT2|BIT0,@PARITY IWRITE NORMAL AND DISABLE
1194 004620 004037 011550          JSR      R0,#CHECKKLOC ICHECK FOR GOOD ABCRT
1195 004624 104003          HLT      +3          IABORTED INCORRECTLY
1196 004626 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 20      TEST DATI (DATA) SM2,DM5 CMP INSTRUCTION
;*****
1200 004632 000004          TST20: SCOPE
;          11/45 **** ROM STATE 177 ****
;
;          11/40 **** ROM STATE 267 ****
1203 004634 004015          JSR      R0,(R5)      ISET UP PARITY VECTOR SERVICE
1205 004636 004706          B5          IROUTINE ADDRESS
1206 004640 011100          MOV      @R1,R0      ISET UP FOR DATO
1207 004642 042777 000004 174752    BIC      #BIT2,@PARITY IWRITE NORMAL
1208 004650 010060          MOV      R0,-2(R0)   IDO THE DATO
1209 004654 052777 000004 174740    BIS      #BIT2,@PARITY IWRITE OTHER PARITY
1210 004662 011110          MOV      @R1,@R0     IDO THE DATO
1211 004664 012737 004674 001332    MOV      #.+10,@#SGDDAT ISTORE THE PC THAT SHOULD
;          IBE PUSHED ON THE STACK
;          IF A PARITY ABORT OCCURS
1214 004672 020050          CMP      R0,@(R0)    IDO A DATI, DATIP
1215 004674 042777 000004 174720    BIC      #BIT2,@PARITY IWRITE NORMAL FOR EMT CALL
1216 004702 104001          HLT      +1          IDIDN'T ABORT
1217 004704 000410          BR       +22         IGO TO NEXT TEST
1218 004706 042777 000005 174706 B5:    BIC      #BIT2|BIT0,@PARITY IWRITE NORMAL AND DISABLE
1219 004714 004037 011550          JSR      R0,#CHECKKLOC ICHECK FOR GOOD ABCRT
1220 004720 104003          HLT      +3          IABORTED INCORRECTLY
1221 004722 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 21      TEST (ADDRESS) SM0,DM5 CMP INSTRUCTION
;*****
1225 004726 000004          TST21: SCOPE
;          11/45 **** ROM STATE 231 ****
;
;          11/40 **** ROM STATE 264 ****
1229 004730 004015          JSR      R0,(R5)      ISET UP PARITY VECTOR SERVICE
1230 004732 004766          B6          IROUTINE ADDRESS
1231 004734 011100          MOV      @R1,R0      ISET UP FOR A DATO
1232 004736 010010          MOV      R0,(R0)     IDO THE DATO
1233 004740 062700 000002          ADD      #2,R0
```

```
1214 004744 012737 004754 001332 MOV #.+10,#$GDDAT ;STORE THE PC THAT SHOULD  
1235 ;BE PUSHED ON THE STACK  
1236 ;IF A PARITY ABORT OCCURS  
Z 1237 004752 020050 CMP R0,0-(R0) ;DO A BATI  
1238 004754 042777 000004 174640 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL  
1239 004762 104001 HLT +1 ;IDIDN'Y ABORT  
1240 004764 000410 BR +22 ;GO TO NEXT TEST  
1241 004766 042777 000005 174626 B61 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE  
1242 004774 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCT  
1243 005000 104003 HLT +3 ;ABORTED INCORRECTLY  
1244 005002 012706 001100 MOV #STACK,SP ;RESET THE STACK  
1245 ;*****  
1246 ;TEST 22 TEST (DATA) SM2,DNO CMP INSTRUCTION  
1247 ;*****  
1248 005006 000004 TST22; SCOPE  
1249 ; 11/45 **** ROM STATE 27 ****  
1250 ;  
1251 ; 11/40 **** ROM STATE 250 ****  
1252 005010 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE  
1253 005012 005042 C ;ROUTINE ADDRESS  
1254 005014 011100 MOV #R1,R0 ;SET UP FOR DATO  
1255 005016 010010 MOV R0,@R0 ;DO THE DATO  
1256 005020 012737 005030 001332 MOV #.+10,#$GDDAT ;STORE THE PC THAT SHOULD  
1257 ;BE PUSHED ON THE STACK  
1258 ;IF A PARITY ABORT OCCURS  
1259 005026 022000 CMP (R0)+,R0 ;DO A BATI  
1260 005030 042777 000004 174564 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL  
1261 005036 104001 HLT +1 ;IDIDN'Y ABORT  
1262 005040 000410 BR +22 ;GO TO NEXT TEST  
1263 005042 042777 000005 174552 Ci BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE  
1264 005050 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCT  
1265 005054 104003 HLT +3 ;ABORTED INCORRECTLY  
1266 005056 012706 001100 MOV #STACK,SP ;RESET THE STACK  
1267 ;*****  
1268 ;TEST 23 TEST (DATA) SM4,DNO CMP INSTRUCTION  
1269 ;*****  
1270 005062 000004 TST23; SCOPE  
1271 ; 11/45 **** ROM STATE 27 ****  
1272 ;  
1273 ; 11/40 **** ROM STATE 250 ****  
1274 005064 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE  
1275 005066 005116 C ;ROUTINE ADDRESS  
1276 005070 011100 MOV #R1,R0 ;SET UP FOR DATO  
Z 1277 005072 010020 MOV R0,(R0)+ ;DO THE DATO
```

```
1278 005074 012737 005104 001332 MOV #.+10,#$GDDAT ;STORE THE PC THAT SHOULD  
1279 ;BE PUSHED ON THE STACK  
1280 ;IF A PARITY ABORT OCCURS  
1281 005102 024000 CMP -(R0),R0 ;DO A BATI  
1282 005104 042777 000004 174510 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL  
1283 005112 104001 HLT +1 ;IDIDN'Y ABORT  
1284 005114 000410 BR +22 ;GO TO NEXT TEST  
1285 005116 042777 000005 174476 C01 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE  
1286 005124 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCT  
1287 005130 104003 HLT +3 ;ABORTED INCORRECTLY  
1288 005132 012706 001100 MOV #STACK,SP ;RESET THE STACK  
1289 ;*****  
1290 ;TEST 24 TEST (DATA) SM3,DNO CMP INSTRUCTION  
1291 ;*****  
1292 005136 000004 TST24; SCOPE  
1293 ; 11/45 **** ROM STATE 146 ****  
1294 ;  
1295 ; 11/40 **** ROM STATE 250 ****  
1296 005140 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE  
1297 005142 005210 C1 ;ROUTINE ADDRESS  
1298 005144 011100 MOV #R1,R0 ;SET UP FOR DATO  
Z 1299 005146 010020 MOV R0,(R0)+ ;DO THE DATO (DATA OTHER PARITY)  
1300 005150 042777 000004 174444 BIC #BIT2,@PARITY ;WRITE NORMAL  
1301 005156 011110 MOV #R1,@R0 ;DO A BATO (ADDRESS NORMAL)  
1302 005160 052777 000004 174434 BIS #BIT2,@PARITY ;WRITE OTHER PARITY  
1303 005166 012737 005176 001332 MOV #.+10,#$GDDAT ;STORE THE PC THAT SHOULD  
1304 ;BE PUSHED ON THE STACK  
1305 ;IF A PARITY ABORT OCCURS  
1306 005174 023000 CMP @R0)+,R0 ;DO A BATI  
1307 005176 042777 000004 174416 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL  
1308 005204 104001 HLT +1 ;IDIDN'Y ABORT  
1309 005206 000410 BR +22 ;GO TO NEXT TEST  
1310 005210 042777 000005 174404 C11 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE  
1311 005216 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCT  
1312 005222 104003 HLT +3 ;ABORTED INCORRECTLY  
1313 005224 012706 001100 MOV #STACK,SP ;RESET THE STACK  
1314 ;*****  
1315 ;TEST 25 TEST (DATA) SM5,DNO CMP INSTRUCTION  
1316 ;*****  
1317 005230 000004 TST25; SCOPE  
1318 ; 11/45 **** ROM STATE 146 ****  
1319 ;  
1320 ; 11/40 **** ROM STATE 250 ****  
1321 005232 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE  
1322 005234 005304 C2 ;ROUTINE ADDRESS  
1323 005236 011100 MOV #R1,R0 ;SET UP FOR A DATO  
1324 005240 042777 000004 174354 BIC #BIT2,@PARITY ;WRITE NORMAL  
1325 005246 010060 MOV R0,-2(R0) ;DO A BATO (ADDRESS NORMAL)  
1326 005252 052777 000004 174342 BIS #BIT2,@PARITY ;WRITE OTHER PARITY  
1327 005260 011110 MOV #R1,@R0 ;DO A BATO (DATA OTHER PARITY)  
1328 005262 012737 005272 001332 MOV #.+10,#$GDDAT ;STORE THE PC THAT SHOULD  
1329 ;BE PUSHED ON THE STACK  
1330 ;IF A PARITY ABORT OCCURS  
1331 ;DO A BATI  
1332 005270 025000 CMP @-(R0),R0 ;DO A BATI
```



```
1332 005272 042777 000004 174322 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1333 005300 104001 HLT +1 ;DIDN'T ABORT
1334 005302 000410 BR +22 ;GO TO NEXT TEST
1335 005304 042777 000005 174310 C21 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1336 005312 004037 011550 JSR R0,@#CHECKLOC ;CHECK FOR GOOD ABCRT
1337 005316 104003 HLT +3 ;ABORTED INCORRECTLY
1338 005320 012706 001100 MOV #STACK,SP ;RESET THE STACK
;*****
;TEST 26 TEST (DATA) SM1,DM0 CMP INSTRUCTION
;*****
1342 005324 000004 TST26: SCOPE
; 11/45 **** ROM STATE 27 ****
;
; 11/40 **** ROM STATE 250 ****
;
1346 005326 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1347 005330 005360 C5 ;ROUTINE ADDRESS
1348 005332 011100 MOV #R1,R0 ;SET UP FOR DATO
1349 005334 010010 MOV R0,#R0 ;DO THE DATO
1350 005336 012737 005346 001332 MOV #,*10,@#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
;IF A PARITY ABORT OCCURS
;DO A BATI
1353 005344 021000 CMP #R0,R0 ;DO A BATI
1354 005346 042777 000004 174246 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1355 005354 104001 HLT +1 ;DIDN'T ABORT
1356 005356 000410 BR +22 ;GO TO NEXT TEST
1357 005360 042777 000005 174234 C31 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1358 005366 004037 011550 JSR R0,@#CHECKLOC ;CHECK FOR GOOD ABCRT
1359 005372 104003 HLT +3 ;ABORTED INCORRECTLY
1360 005374 012706 001100 MOV #STACK,SP ;RESET THE STACK
;*****
;TEST 27 TEST (DATA) SM6,DM0 CMP INSTRUCTION
;*****
1364 005400 000004 TST27: SCOPE
; 11/45 **** ROM STATE 142 ****
;
; 11/40 **** ROM STATE 250 ****
;
1368 005402 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1369 005404 005436 C4 ;ROUTINE ADDRESS
1370 005406 011100 MOV #R1,R0 ;SET UP FOR A DATO
1371 005410 010020 MOV R0,(R0)+ ;DO THE DATO
1372 005412 012737 005424 001332 MOV #,*12,@#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
;IF A PARITY ABORT OCCURS
;DO A BATI
1375 005420 026000 CMP #-2(R0),R0 ;DO A BATI
1376 005424 042777 000004 174170 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1377 005432 104001 HLT +1 ;DIDN'T ABORT
1378 005434 000410 BR +22 ;GO TO NEXT TEST
1379 005436 042777 000005 174156 C41 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1380 005444 004037 011550 JSR R0,@#CHECKLOC ;CHECK FOR GOOD ABCRT
1381 005450 104003 HLT +3 ;ABORTED INCORRECTLY
1382 005452 012706 001100 MOV #STACK,SP ;RESET THE STACK
;*****
;TEST 30 TEST (ADDRESS) SM7,DM0 CMP INSTRUCTION
;*****
1385
1386
```

```
1386 005456 000004 TST30: SCOPE
; 11/45 **** ROM STATE 142 ****
;
; 11/40 **** ROM STATE 245 ****
;
1390 005460 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1391 005462 005514 C5 ;ROUTINE ADDRESS
1392 005464 011100 MOV #R1,R0 ;SET UP FOR DATO
1393 005466 010020 MOV R0,(R0)+ ;DO THE DATO
1394 005470 012737 005502 001332 MOV #,*12,@#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
;IF A PARITY ABORT OCCURS
;DO A BATI
1397 005476 027000 CMP #-2(R0),R0 ;DO A BATI
1398 005502 042777 000004 174112 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1399 005510 104001 HLT +1 ;DIDN'T ABORT
1400 005512 000410 BR +22 ;GO TO NEXT TEST
1401 005514 042777 000005 174100 C51 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1402 005522 004037 011550 JSR R0,@#CHECKLOC ;CHECK FOR GOOD ABCRT
1403 005526 104003 HLT +3 ;ABORTED INCORRECTLY
1404 005530 012706 001100 MOV #STACK,SP ;RESET THE STACK
;*****
;TEST 31 TEST (ADDRESS) SM3,DM0 CMP INSTRUCTION
;*****
1408 005534 000004 TST31: SCOPE
; 11/45 **** ROM STATE 27 ****
;
; 11/40 **** ROM STATE 245 ****
;
1412 005536 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1413 005540 005570 C6 ;ROUTINE ADDRESS
1414 005542 011100 MOV #R1,R0 ;SET UP FOR A DATO
1415 005544 010010 MOV R0,#R0 ;DO THE DATO
1416 005546 012737 005556 001332 MOV #,*10,@#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
;IF A PARITY ABORT OCCURS
;DO A BATI
1419 005554 023000 CMP #(R0),R0 ;DO A BATI
1420 005556 042777 000004 174036 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1421 005564 104001 HLT +1 ;DIDN'T ABORT
1422 005566 000410 BR +22 ;GO TO NEXT TEST
1423 005570 042777 000005 174024 C61 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1424 005576 004037 011550 JSR R0,@#CHECKLOC ;CHECK FOR GOOD ABCRT
1425 005602 104003 HLT +3 ;ABORTED INCORRECTLY
1426 005604 012706 001100 MOV #STACK,SP ;RESET THE STACK
;*****
;TEST 32 TEST (ADDRESS) SM5,DM0 CMP INSTRUCTION
;*****
1430 005610 000004 TST32: SCOPE
; 11/45 **** ROM STATE 27 ****
;
; 11/40 **** ROM STATE 245 ****
;
1434 005612 004015 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1435 005614 005644 C7 ;ROUTINE ADDRESS
1436 005616 011100 MOV #R1,R0 ;SET UP FOR A DATO
1437 005620 010020 MOV R0,(R0)+ ;DO THE DATO
1438 005622 012737 005632 001332 MOV #,*10,@#SGDDAT ;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
1439
```

```

1440      005630 025000      CMP      0-(R0),R0      IIF A PARITY ABORT OCCURS
1441      005632 042777      BIC      #BIT2,@PARITY  IDO A BATI
1442      005640 104001      HLT      +1             IWRITE NORMAL FOR EMT CALL
1443      005642 000410      BR       +2             IDIDN'IT ABORT
1444      005644 042777      BIC      #BIT2|BIT0,@PARITY  IGO TO NEXT TEST
1445      005652 004037      JSR      R0,@#CHECKLOC     IWRITE NORMAL AND CISIBLE
1446      005656 104003      HLT      +3             ICHECK FOR GOOD ABCT
1447      005660 012706      MOV      #STACK,SP        IABORTED INCORRECTLY
1448      005662 012706      MOV      #STACK,SP        IRESET THE STACK
1449      ;
1450      ;
1451      ;
1452      005664 000004      TST33; SCOPE
1453      ;
1454      ;
1455      ;
1456      005666 004015      JSR      R0,(R5)         ISET UP PARITY VECTOR SERVICE
1457      005670 005742      CB       @R1,R0         IROUTINE ADDRESS
1458      005672 011100      MOV      @R1,R0         ISET UP FOR A DATO
1459      005674 042777      BIC      #BIT2,@PARITY  IWRITE NORMAL
1460      005702 010060      MOV      R0,-2(R0)      IDO A BATO (ADDRESS NORMAL)
1461      005706 052777      BIS      #BIT2,@PARITY  IWRITE OTHER PARITY
1462      005714 011110      MOV      @R1,@R0       IDO A DATO (DATA OTHER PARITY)
1463      005716 012737      MOV      #.12,@#SGDDAT  ISTORE THE PC THAT SHOULD
1464      ;
1465      ;
1466      005724 027000      CMP      0-2(R0),R0     IIF A PARITY ABORT OCCURS
1467      005730 042777      BIC      #BIT2,@PARITY  IDO A BATI
1468      005736 104001      HLT      +1             IWRITE NORMAL FOR EMT CALL
1469      005740 000410      BR       +2             IDIDN'IT ABORT
1470      005742 042777      BIC      #BIT2|BIT0,@PARITY  IGO TO NEXT TEST
1471      005750 004037      JSR      R0,@#CHECKLOC     IWRITE NORMAL AND CISIBLE
1472      005754 104003      HLT      +3             ICHECK FOR GOOD ABCT
1473      005756 012706      MOV      #STACK,SP        IABORTED INCORRECTLY
1474      ;
1475      ;
1476      ;
1477      005762 000004      TST34; SCOPE
1478      ;
1479      ;
1480      ;
1481      005764 004015      JSR      R0,(R5)         ISET UP PARITY VECTOR SERVICE
1482      005766 006020      D        @R1,R0         IROUTINE ADDRESS
1483      005770 011100      MOV      @R1,R0         ISET UP FOR A DATO
1484      005772 012710      MOV      @R0,@R0       IDO THE DATO
1485      005776 012737      MOV      #.10,@#SGDDAT  ISTORE THE PC THAT SHOULD
1486      ;
1487      ;
1488      006024 000130      JMP      0-(R0)+        IIF A PARITY ABORT OCCURS
1489      006036 042777      BIC      #BIT2,@PARITY  IDO A BATI
1490      006040 104001      HLT      +1             IWRITE NORMAL FOR EMT CALL
1491      006042 000410      BR       +2             IDIDN'IT ABORT
1492      006044 042777      BIC      #BIT2|BIT0,@PARITY  IGO TO NEXT TEST
1493      006046 004037      JSR      R0,@#CHECKLOC     IWRITE NORMAL AND CISIBLE
1494      006048 011550      HLT      +3             ICHECK FOR GOOD ABCT

```

```

1494      006032 104003      HLT      +3             IABORTED INCORRECTLY
1495      006034 012706      MOV      #STACK,SP        IRESET THE STACK
1496      ;
1497      ;
1498      ;
1499      006040 000004      TST35; SCOPE
1500      ;
1501      ;
1502      ;
1503      006042 004015      JSR      R0,(R5)         ISET UP PARITY VECTOR SERVICE
1504      006044 006076      D        @R1,R0         IROUTINE ADDRESS
1505      006046 011100      MOV      @R1,R0         ISET UP FOR A DATO
1506      006050 012720      MOV      @R0,(R0)+      IDO THE DATO
1507      006054 012737      MOV      #.10,@#SGDDAT  ISTORE THE PC THAT SHOULD
1508      ;
1509      ;
1510      006062 000130      JMP      0-(R0)         IIF A PARITY ABORT OCCURS
1511      006064 042777      BIC      #BIT2,@PARITY  IDO A BATI
1512      006066 104001      HLT      +1             IWRITE NORMAL FOR EMT CALL
1513      006068 000410      BR       +2             IDIDN'IT ABORT
1514      006070 042777      BIC      #BIT2|BIT0,@PARITY  IGO TO NEXT TEST
1515      006072 004037      JSR      R0,@#CHECKLOC     IWRITE NORMAL AND CISIBLE
1516      006074 004037      JSR      R0,@#CHECKLOC     ICHECK FOR GOOD ABCT
1517      006076 104003      HLT      +3             IABORTED INCORRECTLY
1518      006078 012706      MOV      #STACK,SP        IRESET THE STACK
1519      ;
1520      ;
1521      ;
1522      006116 000004      TST36; SCOPE
1523      ;
1524      ;
1525      ;
1526      006120 004015      JSR      R0,(R5)         ISET UP PARITY VECTOR SERVICE
1527      006122 006156      D        @R1,R0         IROUTINE ADDRESS
1528      006124 011100      MOV      @R1,R0         ISET UP FOR A DATO
1529      006126 012720      MOV      @R0,(R0)+      IDO THE DATO
1530      006128 012737      MOV      #.12,@#SGDDAT  ISTORE THE PC THAT SHOULD
1531      ;
1532      ;
1533      006140 000170      JMP      0-2(R0)        IIF A PARITY ABORT OCCURS
1534      006142 042777      BIC      #BIT2,@PARITY  IDO A BATI
1535      006144 104001      HLT      +1             IWRITE NORMAL FOR EMT CALL
1536      006146 000410      BR       +2             IDIDN'IT ABORT
1537      006148 042777      BIC      #BIT2|BIT0,@PARITY  IGO TO NEXT TEST
1538      006150 004037      JSR      R0,@#CHECKLOC     IWRITE NORMAL AND CISIBLE
1539      006152 104003      HLT      +3             ICHECK FOR GOOD ABCT
1540      006154 012706      MOV      #STACK,SP        IABORTED INCORRECTLY
1541      ;

```

THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:

1ST PUSH - ADDRESS OF THE TAG 'VEGSET' (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

1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601

;TEST 37 TEST MARK "POP" MARK INSTRUCTION
;*****
TST37: SCOPE 11/45 **** ROM STATE 260 ****
;
; 11/40 **** ROM STATE 1 ****
JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
E0 IROUTINE ADDRESS
BIC #BIT2,@PARITY IWRITE 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
;
;*****

1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653

;*****
;*****
MOV #NEWSTK,SP ISET THE STACK IN PARITY MEMORY AREA
MOV #INTERAD,R2 IGET THE INTERLEAVE FACTOR FOR THIS CONTROLLER
DEC R2 ICALCULATE NO. OF PARAMETERS TO BE PUSHED ON THE STACK
MOV R5,=(SP) IPUSH #B CONTENTS ON STACK
TST R2 IANY PARAMETERS TO BE PUSHED ON THE STACK?
BEQ Z\$ IBRANCH IF NO
DEC R2 ISUBTRACT 1 FROM PARAMETER COUNT
MOV #1,-(SP) IPUSH #PARAMETER ONTO STACK
BR IS IGO BACK TO SEE IF ANY MORE!
BIS #BIT2,@PARITY IWRITE OTHER PARITY
MOV #INTERAD,R2 IGET THE INTERLEAVE FACTOR FOR THIS CONTROLLER
DEC R2 ICALCULATE NO. OF PARAMETERS THAT WERE TO BE PUSHED ON THE STACK
ADD #6400,R2 ICALCULATE THE CORRESPONDING MARK INSTRUCTION
MOV R2,=(SP) IPUSH MARK INSTRUCTION ON STACK
BIC #BIT2,@PARITY IWRITE NORMAL
MOV SP,R3 IPLACE MARK INSTRUCTION ADDRESS INTO R3 FOR RTS
MOV SP,R2 IGET THE PC OF THE MARK INSTRUCTION
TST #CPU40 IARE WE ON AN 11/40?
BNE Z\$ IBRANCH IF YES
AND DON'T STEP UP THE PC IIT WON'T BE UPDATED ON THE PARITY ABORT
ADD #2,R2 ISTEP UP THE PC IIT WILL BE UPDATED ON THE PARITY ABORT
MOV R2,@SGDDAT ISTORE THIS VALUE OF PC ITHAT SHOULD BE PUSHED ON THE STACK IF A PARITY ABORT OCCURS
JSR PC,MRK0 ISUBROUTINE CALL
HLT +1 IDON'T ABORT
BR +2 IGO TO RESET THE STACK
RTS R5 IRETURN FROM SUBROUTINE
BIC #BIT0,@PARITY IDISABLE PARITY
JSR R0,@CHECKLOC ICHECK FOR GOOD ABORT
HLT +3 IABORTED INCORRECTLY
MOV #NEWSTK,SP IRESET THE STACK
;*****

```

|*****|
|*****|
|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|
|          |
|INTH PUSH|
|INTH +1 PUSH - MARK INSTRUCTION (NORMAL)|
|LAST PUSH - OLD PC FROM THE 'JSR' (NORMAL)|
|NOTE1 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|
|          |
|*****|
|*****|
|*****|

```

```

1680
1689
1690
1691 006346 012705 011450      MOV    #VECSET,R5      IRESTORE THE PARITY VECTOR
1692                                     ISERVICE ADDRESS SETUP ROUTINE
1693                                     IADDRESS
1694
1695                                     I*****
1696 ITEST 40      TEST    OLD REGISTER CONTENTS MARK INSTRUCTION
1697 006352 000004      TST40I SCOPE
1698                                     I          11/45 **** ROM STATE 235 ****
1699                                     I
1700                                     I
1701                                     I          11/40 **** ROM STATE 356 ****
1702 006354 004015      JSR    R0,(R5)      ISET UP PARITY VECTOR SERVICE
1703 006356 006510      E2                                     IROUTINE ADDRESS
1704 006360 017702 173062      MOV    #NTERAD,R2   IGET THE INTERLEAVE FACTOR
1705                                     IFOR THIS CONTROLLER
1706 006364 005302      DEC    R2           ICALCULATE NO. OF PARAMETERS
1707                                     ITO BE PUSHED ON THE STACK
1708 006366 042777 000004 173226      BIC    #BIT2,#PARITY IWRITE NORMAL
1709 006374 005737 001642      TST    #CPU40      IARE WE ON AN 11/40?
1710 006400 001407      BEQ    35          IBRANCH IF NO

```

```

1710
1711
1712 006402 013700 001476      MOV    #NEWSTK,R0   IDATA WILL BE TAKEN FIRST AND
1713 006406 162700 000002      SUB    #2,R0        ITHEN THE PC UPDATED
1714                                     IGET THE INITIAL STACK POINT
1715                                     IDROP DOWN 1 WORD ADDRESS
1716 006412 010037 001332      MOV    R0,#SGODAT  IBECAUSE THE PC WILL BE UPDATED
1717                                     IFIRST THEN THE DATA TAKEN
1718                                     ISTORE THIS VALUE AS THE PC
1719                                     ITHAT SHOULD BE PUSHED ON THE STACK
1720 006416 000403      BR     45          IIF A PARITY ABORT OCCURS
1721 006420 012737 006502 001332 35I  MOV    #E1,#SGODAT ICONTINUE WITH TEST
1722                                     ISTORE THE PC THAT SHOULD
1723 006426 052777 000004 173166 45I  BIS    #BIT2,#PARITY IBE PUSHED ON THE STACK
1724 006434 010546      MOV    R5,-(SP)    IIF A PARITY ABORT OCCURS
1725 006436 042777 000004 173156      BIC    #BIT2,#PARITY IWRITE OTHER PARITY
1726 006444 005702 15I  TST    R2           ISTORE OLD R5 CONTENTS ON STACK
1727                                     IWRITE NORMAL
1728 006446 001404      BEQ    25          IANY PARAMETERS TO BE PUSHED
1729 006450 005302      DEC    R2           ION THE STACK?
1730 006452 012746 000001      MOV    #1,-(SP)    IBRANCH IF NO
1731 006456 000772      BR     15          ISUBTRACT 1 FROM PARAMETER COUNT
1732 006460 017702 172762 25I  MOV    #NTERAD,R2  IPUSH PARAMETER ON STACK
1733                                     IGO BACK TO SEE IF ANY MORE!
1734 006464 005302      DEC    R2           IGET THE INTERLEAVE FACTOR
1735                                     IFOR THIS CONTROLLER
1736                                     ICALCULATE NO. OF PARAMETERS
1737 006466 062702 006400      ADD    #6400,R2    ITHAT WERE TO BE PLSHD ON THE
1738                                     ISTACK
1739 006472 010246      MOV    R2,-(SP)    ICALCULATE THE CORRESPONDING
1740 006474 010605      MOV    SP,R5       IMARK INSTRUCTION
1741                                     IPUSH MARK INSTRUCTION ON STACK
1742 006476 004767 000004      JSR    PC,MRK1     IPLACE MARK INSTRUCTION ADDRESS
1743 006502 104001      E1I  HLT    +1       INTO R5 FOR RTS
1744 006504 000407      BR     +20        ISUBROUTINE CALL
1745 006506 000205      MRK1I RTS    R5     IDON'T ABORT
1746 006510 042777 000001 173104      BIC    #BIT0,#PARITY IGO TO RESET THE STACK
1747 006516 004037 011550      JSR    R0,#CHECKLOC IRETURN FROM SUBROUTINE
1748 006522 104003      HLT    +3         IDISABLE PARITY
1749 006524 013706 001476      MOV    #NEWSTK,SP  ICHECK FOR GOOD ABCRT
1750 006530 012705 011450      MOV    #VECSET,R5  IABORTED INCORRECTLY
1751                                     IRESET THE STACK
1752                                     IRESTORE THE PARITY VECTOR
1753                                     ISERVICE ADDRESS SETUP ROUTINE
1754                                     IADDRESS
1755
1756 ITEST 41      TEST    SOB BRANCH SOB INSTRUCTION
1757 006534 000004      TST41I SCOPE
1758                                     I          11/45 **** ROM STATE 240 ****
1759                                     I
1760                                     I          11/40 **** ROM STATE 1 ****
1761 006536 004015      JSR    R0,(R5)     ISET UP PARITY VECTOR SERVICE
1762 006540 006644      F0                                     IROUTINE ADDRESS
1763 006542 042777 000004 173052      BIC    #BIT2,#PARITY IWRITE NORMAL
1764 006550 012700 000004      MOV    #4,R0       IMOVE A NUMBER >1 TO R0 SO THAT

```

1764
1765
1766 006554 011102 MOV R1,R2
1767 006556 013703 001476 MOV R0,NEWSTK,R3
1768 006562 005737 001642 TST R0,CPU40
1769 006566 001403 BEQ 1\$
1770 006570 062703 000002 ADD #2,R3
1771
1772
1773 006574 000402 BR 2\$
1774 006576 062703 000004 1\$ ADD #4,R3
1775
1776
1777 006602 010337 001332 2\$ MOV R3,MSGDDAT
1778
1779
1780 006606 052777 000004 173006 BJS #BIT2,PARITY
1781 006614 512722 000240 MOV #240,(R2)*
1782
1783
1784
1785
1786 006620 042777 000004 172774 BIC #BIT2,PARITY
1787 006626 012712 000203 MOV #203,R2
1788
1789
1790
1791 006632 000403 BR +10
1792 006634 004342 F1 JSR R3,-(R2)
1793
1794
1795 006636 104001 HLT +1
1796 006640 000407 BR +20
1797 006642 077004 SOB R0,F
1798 006644 042777 000001 172750 F0 BIC #BIT0,PARITY
1799 006652 004037 011550 JSR R0,MCHECKLOC
1800 006656 104003 HLT +3
1801 006660 013706 001476 MOV R0,NEWSTK,SP
1802
1803
1804
1805 006664 000004
1806
1807
1808
1809
1810 006666 004015 JSR R0,(R5)
1811 006670 006774 F1
1812 006672 011100 MOV R1,R0
1813 006674 017702 172546 MOV R0,ENTERAD,R2
1814
1815 006700 042777 000004 172714 BIC #BIT2,PARITY
1816 006706 005702 1\$ TST R2
1817 006710 001404 BEQ 2\$

WHEN DECREMENTED BY "SOB" THE
RESULT WON'T BE 0
ISET UP FOR A CATO
IGET THE INITIAL TEST POINT
IARE WE ON AN 11/40?
IBRANCH IF NO
ISTEP UP 1 WORD
ISINCE THE PC WILL NOT BE
IUPDATED ON THE PARITY ABORT
IGO TO STORE R1'S VALUE
ISTEP UP 2 WORDS
ISINCE THE PC WILL BE UPDATED
ION THE PARITY ABORT
ISTORE THIS VALUE CT PC
ITHAT SHOULD BE PUSHED ON THE
ISTACK IF A PARITY ABORT OCCURS
IWRITE OTHER PARITY
IINSTRUCTION TO BE DONE ON
ICOMPLETION OF SOB INSTRUCTION
ITHE INSTRUCTION WILL BE IN
IPARITY MEMORY AS DESIGNATED BY
ITHE CONTENTS OF R1
IWRITE NORMAL
IWRITE AN RTS R3 INTO THE LOCATION
IFOLLOWING THE INSTRUCTION TO BE
IEXECUTED IN PARITY MEMORY IN
ICASE IT DOESN'T ABORT
IGO TO SOB INSTRUCTION
ITHE INSTRUCTION TO BE DONE ON
ICOMPLETION OF SOB EXECUTION IS
IPLACED HERE
IDIDN'T ABORT
IGO TO RESET THE STACK
IEXECUTE THIS INSTRUCTION
IDISABLE PARITY
ICHECK FOR GOOD ABRCT
IABORTED INCORRECTLY
IRESET THE STACK

I TEST 42 TEST SH0,DH0 MOV INSTRUCTION

I TST42: SCOPE
I
I 11/40 **** ROM STATE 20 ****
I
I
I 11/40 **** ROM STATE 1 ****
I
I
I
I
I

1818 006712 005302 DEC R2
1819
1820 006714 012720 010000 MOV #010000,(R0)*
1821
1822 006720 000772 BR 1\$
1823 006722 052777 000004 172672 2\$ BJS #BIT2,PARITY
1824 006730 012720 010000 MOV #010000,(R0)*
1825
1826
1827 006734 042777 000004 172640 BIC #BIT2,PARITY
1828 006742 012710 000203 MOV #203,(R0)
1829
1830
1831
1832 006746 005737 001642 TST R0,CPU40
1833 006752 001402 BEQ 3\$
1834
1835
1836 006754 102700 000002 SUB #2,R0
1837
1838
1839 006760 010037 001332 3\$ MOV R0,MSGDDAT
1840
1841
1842 006764 004371 000000 JSR R3,(R1)
1843 006770 104001 HLT +1
1844 006772 000406 BR +16
1845 006774 042777 000001 172620 F1 BIC #BIT0,PARITY
1846 007002 004037 011550 JSR R0,MCHECKLOC
1847 007006 104003 HLT +3
1848 007010 013706 001476 MOV R0,NEWSTK,SP
1849
1850
1851

;SUBTRACT 1 FROM INSTRUCTION
ICOUNT
IMOVE THE INSTRUCTION 'MOV R0,R0'
ITO PARITY AREA
IGO BACK TO SEE IF ANY MORE;
IWRITE OTHER PARITY
IMOVE THE INSTRUCTION 'MOV R0,R0'
IINTO THE NEXT WORD LOCATION AFTER
ITHE PREVIOUS 'MOV R0,R0' INSTRUCTION
IWRITE NORMAL
IMOVE RTS R3 INTO NEXT WORD
ILOCAION AFTER THE PREVIOUS
IMOV R0,R0' INSTRUCTION IN CASE
IA PARITY ABORT DOESN'T OCCUR
IARE WE ON AN 11/40?
IBRANCH IF NO
ISINCE PC WILL BE UPDATED ON THE
IPARITY ABORT
IDROP BACK 1 WORD ADDRESS
ISINCE THE PC WILL NOT BE
IUPDATED ON THE PARITY ABORT
ISTORE THE PC THAT SHOULD
IBE PUSHED ON THE STACK
IIF A PARITY ABORT OCCURS
IGO TO PARITY MEMORY AREA
IDIDN'T ABORT
IGO TO RESET THE STACK
IDISABLE PARITY
ICHECK FOR GOOD ABRCT
IABORTED INCORRECTLY
IRESET THE STACK

I
I THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS I
I
I 1ST PUSH - OLD R3 FROM THE 'JSR' (OTHER PARITY)
I
I NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
I R3 CONTENTS (1ST PUSH)
I
I
I
I
I

```
1872 ;*****
1873 ;TEST 43 TEST RTS INSTRUCTION
1874 ;*****
1875 007014 000004 TST43i SCOPE
1876 ;
1877 ; 11/45 *** ROM STATE 224 ***
1878 ;
1879 ; 11/40 *** ROM STATE 325 ***
1880 007016 004015 JSR R0,(R0) ;SET UP PARITY VECTOR SERVICE
1881 007020 007066 G0 ;ROUTINE ADDRESS
1882 007022 042777 000004 172572 BIC #BIT2,0PARITY ;WRITE NORMAL
1883 007030 012700 007056 001332 MOV #G,R0 ;SET UP A SUBROUTINE ADDRESS
1884 007034 012737 007052 001332 MOV #.16,0$GDDAT ;STORE THE PC THAT SHOULD
1885 ; ; BE PUSHED ON THE STACK
1886 007042 052777 000004 172552 BIS #BIT2,0PARITY ;IF A PARITY ABORT OCCURS
1887 007050 004320 JSR R3,(R0)+ ;WRITE OTHER PARITY
1888 007052 104001 HLT +1 ;SUBROUTINE CALL
1889 007054 000412 BR .+26 ;DIDN'T ABORT
1890 007056 042777 000004 172536 Gi BIC #BIT2,0PARITY ;GO TO RESET THE STACK
1891 007064 000203 RTS R3 ;WRITE NORMAL
1892 007066 042777 000001 172526 G0i BIC #BIT0,0PARITY ;RETURN FROM SUBROUTINE
1893 007074 004037 011550 JSR R0,0$CHECKLOC ;DISABLE PARITY
1894 007100 104003 HLT +3 ;CHECK FOR GOOD ABCRT
1895 007102 013706 001476 MOV 0$NEWSTK,SP ;ABORTED INCORRECTLY
1896 ; ; IRESET THE STACK
1897
1898
```

```
*****
*****
*****
;
; 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'
;
;*****
;*****
```

```
1928 ;*****
1929 ;*****
1930 ;TEST 44 TEST 1ST POP RTI INSTRUCTION
1931 ;*****
1932 TST44i SCOPE
1933 007106 000004 ;
1934 ; 11/45 *** ROM STATE 212 ***
1935 ;
1936 ; 11/40 *** ROM STATE 320 ***
1937 007110 004015 JSR R0,(R0) ;SET UP PARITY VECTOR SERVICE
1938 007112 007150 H0 ;ROUTINE ADDRESS
1939 007114 012746 000340 MOV #340,-(SP) ;SET PS FOR 'RTI'
1940 007120 012746 007134 MOV #H1,-(SP) ;SET RETURN FROM 'RTI'
1941 007124 012737 007134 001332 MOV #H1,0$GDDAT ;STORE THE PC THAT SHOULD
1942 ; ; BE PUSHED ON THE STACK
1943 ; ; IF A PARITY ABORT OCCURS
1944 007132 004002 BR H ;GO DO 'RTI'
1945 007134 104001 HLT +1 ;DIDN'T ABORT
1946 007136 000412 BR .+26 ;GO TO RESET THE STACK
1947 007140 042777 000004 172454 Hi BIC #BIT2,0PARITY ;WRITE NORMAL
1948 007146 000002 RTS ;RETURN FROM INTERRUPT
1949 007150 042777 000001 172444 H0i BIC #BIT0,0PARITY ;DISABLE PARITY
1950 007156 004037 011550 JSR R0,0$CHECKLOC ;CHECK FOR GOOD ABCRT
1951 007162 104003 HLT +3 ;ABORTED INCORRECTLY
1952 007164 013706 001476 MOV 0$NEWSTK,SP ;RESET THE STACK
1953
1954
1955
```

```
*****
*****
*****
;
; 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)
```

```

;
;*****
;*****
;*****
1986
1987
1988
1989 ;***** TEST 45 TEST 2ND POP RTI INSTRUCTION *****
1990 ;***** TST45: SCOPE *****
1991 ;***** 11/45 **** ROM STATE 214 **** *****
1992 ;
1993 ;
1994 ; 11/40 **** ROM STATE 322 ****
1995 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
1996 M2 ;ROUTINE ADDRESS
1997 MOV #340,-(SP) ;SET PS FOR 'RTI'
1998 MOV #H3,-(SP) ;SET RETURN FROM 'RTI'
1999 MOV #H3,#SGDDAT ;STORE THE PC THAT SHOULD
2000 ;BE PUSHED ON THE STACK
2001 ;IF A PARITY ABORT OCCURS
2002 BR W4 ;GO DO 'RTI'
2003 H3: HLT +1 ;IDONT' ABORT
2004 BR +36 ;GO TO RESET THE STACK
2005 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE PARITY
2006 H4: MOV (SP),-(SP) ;WRITE 1ST 'POPI' NORMAL SO NO
2007 ;ERROR WILL OCCUR
2008 BIS #BIT0,@PARITY ;ENABLE PARITY
2009 RTI ;RETURN FROM INTERRUPT
2010 BIC #BIT0,@PARITY ;DISABLE PARITY
2011 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT
2012 H2: HLT +3 ;ABORTED INCORRECTLY
2013 MOV #STACK,SP ;INITIALIZE THE STACK BACK
2014 ;TO 1100 FOR WE DON'T NEED IT
2015 ;TO BE IN PARITY MEMORY AREA
2016 ;ANYMORE
2017 ;*****
2018 ;*****
2019 ;***** TEST 46 TEST (DATA) SM6,DH0 ADD INSTRUCTION *****
2020 ;***** TST46: SCOPE *****
2021 ;***** 11/45 **** ROM STATE 142 **** *****
2022 ;
2023 ;
2024 ; 11/40 **** ROM STATE 250 ****
2025 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
2026 L ;ROUTINE ADDRESS
2027 MOV #R1,R0 ;SET UP FOR A DATO
2028 MOV R0,(R0) ;DO THE DATO
2029 Z: MOV #,+12,#SGDDAT ;STORE THE PC THAT SHOULD
2030 ;BE PUSHED ON THE STACK
2031 ;IF A PARITY ABORT OCCURS
2032 ADD -2(R0),R0 ;DO A DATI
2033 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2034 HLT +1 ;IDONT' ABORT

```

```

2034 BR +22 ;GO TO NEXT TEST
2035 BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2036 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT
2037 HLT +3 ;ABORTED INCORRECTLY
2038 MOV #STACK,SP ;RESET THE STACK
2039 ;*****
2040 ;*****
2041 ;***** TEST 47 TEST DM1 TSTB INSTRUCTION *****
2042 ;***** TST47: SCOPE *****
2043 ;***** 11/45 **** ROM STATE 175 **** *****
2044 ;
2045 ;
2046 ; 11/40 **** ROM STATE 267 ****
2047 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
2048 M ;ROUTINE ADDRESS
2049 MOV #R1,R0 ;SET UP FOR A DATO
2050 MOV R0,(R0) ;DO THE DATO
2051 MOV #,+10,#SGDDAT ;STORE THE PC THAT SHOULD
2052 ;BE PUSHED ON THE STACK
2053 ;IF A PARITY ABORT OCCURS
2054 TSTB (R0) ;DO A DATI, DATIP
2055 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2056 HLT +1 ;IDONT' ABORT
2057 BR +22 ;GO TO NEXT TEST
2058 H4: BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2059 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT
2060 HLT +3 ;ABORTED INCORRECTLY
2061 MOV #STACK,SP ;RESET THE STACK
2062 ;*****
2063 ;*****
2064 ;***** TEST 50 TEST Dh3 CLR B INSTRUCTION *****
2065 ;***** TST50: SCOPE *****
2066 ;***** 11/45 **** ROM STATE 221 **** *****
2067 ;
2068 ;
2069 ; 11/40 **** ROM STATE 264 ****
2070 JSR R0,(R5) ;SET UP PARITY VECTOR SERVICE
2071 NN ;ROUTINE ADDRESS
2072 MOV #R1,R0 ;SET UP FOR A DATO
2073 MOV R0,(R0) ;DO THE DATO
2074 MOV #,+10,#SGDDAT ;STORE THE PC THAT SHOULD
2075 ;BE PUSHED ON THE STACK
2076 ;IF A PARITY ABORT OCCURS
2077 CLR B (R0) ;DO A DATI
2078 HLT +1 ;IDONT' ABORT
2079 BR +22 ;GO TO NEXT TEST
2080 H4: BIC #BIT2|BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2081 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT
2082 HLT +3 ;ABORTED INCORRECTLY
2083 MOV #STACK,SP ;RESET THE STACK
2084 ;*****
2085 ;*****
2086 ;***** TEST 51 TEST SH1 SUBTRACT INSTRUCTION *****
2087 ;***** TST51: SCOPE *****
2088 ;***** 11/45 **** ROM STATE 27 **** *****
2089 ;

```

```

2088 ;
2089 ;
2090 007472 004015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2091 007474 007526 P IROUTINE ADDRESS
2092 007476 011100 MOV #R1,R0 ISET UP FOR A DATO
2093 007500 010010 MOV R0,(R0) IDO THE DATO
2094 007502 012737 007512 001332 MOV #.+10,#SGDDAT ISTORE THE PC THAT SHOULD
2095 IBE PUSHED ON THE STACK
2096 IF A PARITY ABORT OCCURS
2097 007510 161027 177777 SUB (R0),#177777 IDO A BATI
2098 007514 042777 000004 172100 BIC #BIT2,@PARITY IWRITE NORMAL FOR EMT CALL
2099 007522 104001 HLT +1 IDIDN'T ABORT
2100 007524 000410 BR .+22 IGO TO NEXT TEST
2101 007526 042777 000005 172066 PI BIC #BIT2|BIT0,@PARITY IWRITE NORMAL AND CISIBLE
2102 007534 004037 011550 JSR R0,@CHECKLOC ICHECK FOR GOOD ABCRT
2103 007540 104003 HLT +3 IABORTED INCORRECTLY
2104 007542 012706 001100 MOV #STACK,SP IRESET THE STACK
;*****
;TEST 52 TEST (DATA) SMB B1SB INSTRUCTION
;*****
TST52| SCOPE
;
; 11/45 **** ROM STATE 177 ****
;
;
; 11/40 **** ROM STATE 245 ****
;
;
2112 007550 004015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2113 007552 007604 R IROUTINE ADDRESS
2114 007554 011100 MOV #R1,R0 ISET UP FOR A DATO
2115 007556 010010 MOV R0,(R0) IDO THE DATO
2116 007560 012737 007570 001332 MOV #.+10,#SGDDAT ISTORE THE PC THAT SHOULD
2117 IBE PUSHED ON THE STACK
2118 IF A PARITY ABORT OCCURS
2119 007566 155060 000002 B;SB 0-(R0),2(R0) IDO A BATI
2120 007572 042777 000004 172022 BIC #BIT2,@PARITY IWRITE NORMAL FOR EMT CALL
2121 007600 104001 HLT +1 IDIDN'T ABORT
2122 007602 000410 BR .+22 IGO TO NEXT TEST
2123 007604 042777 000005 172010 RI BIC #BIT2|BIT0,@PARITY IWRITE NORMAL AND CISIBLE
2124 007612 004037 011550 JSR R0,@CHECKLOC ICHECK FOR GOOD ABCRT
2125 007616 104003 HLT +3 IABORTED INCORRECTLY
2126 007620 012706 001100 MOV #STACK,SP IRESET THE STACK
;*****
;TEST 53 NEXT 4 TESTS ARE NON-INTERLEAVE 4K DEPENDENT
;*****
TST53| SCOPE
;
; CMP #17360,R1 IHAVE WE PARITY IN THE LOWER 4K?
; AND IS THE SELECTED REGISTER
; GOVERNING THE 4K AREA?
; THIS COMPARE IS IF THE KT11
; ISN'T ENABLED DURING PROGRAM
; EXECUTION
; YES - PROCEED TO NEXT TESTS
; WHICH DEPEND ON PARITY IN LOWER 4K
; THE ABOVE COMPARE DIDN'T
; CHECK - - DO THIS COMPARE TO
; SEE IF IT WAS BECAUSE THE

```

```

2142 ;
2143 ;
2144 007642 001401 BEQ 25 ;TABLE WAS CREATED WITH MEMORY
;MANAGEMENT TURNED ON
;PROCEED TO NEXT TESTS IF THIS
;COMPARE CHECKS
2145 ;
2146 007644 000404 BR 45 ;THE REGISTER UNDER TEST IS NOT
;CONTROLING THE LOWER 4K!!!
;GO TO RE-EVALUATE STSTNM AND
;JUMP OVER THE (4) 4K DEPENDENT
;TESTS
2147 ;
2148 ;
2149 ;
2150 ;
2151 007646 022777 000001 171572 25; CMP #1,@ENTERAD ;IS THIS CONTROLLER
;INTERLEAVED??
2152 ;
2153 007654 001405 BEQ 35 ;BRANCH IF NO
2154 007656 062767 000004 171416 45; ADD #4,STSTNM ;SET STSTNM TO PROPER VALUE
;SINCE THE NEXT 4 TESTS WILL
;BE SKIPPED
2155 ;
2156 ;
2157 007664 000137 010426 JMP #RED+26 ;JUMP TO 1ST INDEX WORD TEST
2158 007670 005237 001624 35; INC #PS@CORZONES ;SET FLAG INDICATING TO 'CHECKLOC'
;ROUTINE THAT PS AND PC FETCH
;AND RED & YELLOW ZONE AREAS
;ARE GOING TO BE TESTED
;*****
;
;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
;
;*****
;*****
;*****
;*****
;*****
;*****
2190 ;
2191 ;
2192 ;
2193 ;
2194 ;
2195 007674 000004 TST54| SCOPE ;TEST 54 TEST NEW 'ps' FETCH
;*****

```



```

2196 ; ; 11/45 **** ROM STATE 397 ****
2197 ; ;
2198 ; ; 11/40 **** ROM STATE 115 ****
2199 007676 004015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2200 007700 007754 Y0 IROUTINE ADDRESS
2201 007702 042777 000004 171712 BIC #BIT2,#PARITY IWRITE NORMAL
2202 007710 012737 007750 000010 MOV #T,#RESVEC IPRESERVED INSTRUCTION TIMEOUT
2203 IVECTOR ADDRESS
2204 007716 012737 007750 001332 MOV #T,#SGDDAT ISTORE THE PC THAT SHOULD
2205 IBE PUSHED ON THE STACK
2206 IIF A PARITY ABORT OCCURS
2207 007724 052777 000004 171670 BIS #BIT2,#PARITY IWRITE OTHER PARITY
2208 007732 012737 000340 000012 MOV #340,#RESVEC+2 INEW 'PS' FOR ERROR TRAP
2209 007740 042777 000004 171654 BIC #BIT2,#PARITY IWRITE NORMAL
2210 007746 007000 7000 INON-RECOGNIZABLE CP-CODE
2211 I SHOULD ATTEMPT TO TRAP TO 'TI'
2212 007750 104001 TI HLT +1 IDIDN'T ABORT
2213 007752 000406 BR .+16 IGO TO RESET THE STACK
2214 007754 042777 000001 171640 T01 BIC #BIT0,#PARITY IDISABLE PARITY
2215 007742 004037 011550 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2216 007746 104003 HLT +3 IABORTED INCORRECTLY
2217 007770 012706 001100 MOV #STACK,SP IRESET THE STACK
2218
2219
2220

```

```

|*****|
|*****|
|*****|
|*****|
|*****|

```

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!!

```

```

|*****|
|*****|
|*****|
|*****|
|*****|

```

2248
2249

```

2250 ;*****|
2251 ;TEST 95 TEST NEW 'PC' FETCH
2252 ;*****|
2253 007774 000004 TST55| SCOPE
2254 ; ; 11/45 **** ROM STATE 395 ****
2255 ; ;
2256 ; ; 11/40 **** ROM STATE 333 ****
2257 007776 004015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2258 010000 010056 Y0 IROUTINE ADDRESS
2259 010002 012737 010052 000010 MOV #V,#RESVEC IPRESERVED INSTRUCTION TIMEOUT
2260 IVECTOR ADDRESS
2261 010010 012737 000340 000012 MOV #340,#RESVEC+2 INEW 'PS' FOR ERROR TRAP
2262 010016 005737 001642 TST #CPU40 IARE WE ON AN 11/42?
2263 010022 001404 BEQ 1$ IBRANCH IF NO
2264 010024 012737 010052 001332 MOV #V,#SGDDAT ISTORE THIS VALUE OF THE PC THAT
2265 I SHOULD BE PUSHED ON THE STACK
2266 IIF A PARITY ABORT OCCURS
2267 ISINCE THE PC IS UPDATED FIRST
2268 ITHEN THE VECTOR DATA TAKEN
2269 010032 000403 BR 2$ ICONTINUE WITH TEST
2270 010034 012737 000010 001332 1$1 MOV #10,#SGDDAT ISTORE THE PC THAT SHOULD
2271 IBE PUSHED ON THE STACK
2272 IIF A PARITY ABORT OCCURS
2273 010042 042777 000004 171552 2$1 BIC #BIT2,#PARITY IWRITE NORMAL
2274 010050 007000 7000 INON-RECOGNIZABLE CP-CODE
2275 I SHOULD ATTEMPT TO TRAP TO 'VI'
2276 010052 104001 VI HLT +1 IDIDN'T ABORT
2277 010054 000406 BR .+16 IGO TO RESET THE STACK
2278 010056 042777 000001 171536 V01 BIC #BIT0,#PARITY IDISABLE PARITY
2279 010064 004037 011550 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2280 010070 104003 HLT +3 IABORTED INCORRECTLY
2281 010072 012706 001100 MOV #STACK,SP IRESET THE STACK
2282
2283
2284

```

```

|*****|
|*****|
|*****|
|*****|
|*****|

```

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!
 ;

```

2318
2319
2320
2321 ;TEST 26 TEST ABORT IN 'YELLOW' ZONE
2322 ;*****
2323 010076 000004 ;TST56: SCOPE
2324 ;
2325 ; 11/45 **** ROM STATE 177 ****
2326 ;
2327 010100 004015 ; 11/40 **** ROM STATE 267 ****
2328 010102 010102 JSR R0,(R0) ;SET UP PARITY VECTOR SERVICE
2329 010104 005037 000340 CLR #340 ;ROUTINE ADDRESS
2330 ; ;CLEAR BOTTOM LIMIT LOCATION
2331 ; ;OF 'YELLOW' ZONE USING 'OTHER'
2332 ; ;IPARITY
2333 010110 042777 000004 171504 BIC #BIT2,0PARITY ;WRITE NORMAL
2334 010112 012706 000376 MOV #376,SP ;SET STACK IN 'YELLOW' AREA
2335 ; ;SAVE CONTENTS OF LOC, 4 IN
2336 010126 000000 SAVLOC4: .WORD 0 ;NEXT LOCATION
2337 ; ;ORIGINAL CONTENTS OF LOC, 4
2338 010130 012737 010166 000004 MOV #V2,0ERRVEC ;GO HERE
2339 ; ;SET UP A TIMEOUT VECTOR SERVICE
2340 ; ;ROUTINE ADDRESS FOR STACK VIO-
2341 ; ;LATION
2342 010136 012737 000340 000006 MOV #340,0ERRVEC-2 ;NEW PS ON TIMEOUT TRAP
2343 010144 012737 010156 001332 MOV #.12,0NSGDAT ;STORE THE PC THAT SHOULD
2344 ; ;BE PUSHED ON THE STACK
2345 ; ;IF A PARITY ABORT OCCURS
2346 ; ;PREFERENCE BOTTOM LIMIT OF
2347 ; ;'YELLOW' ZONE USING REGISTER 6
2348 ; ;THIS INSTRUCTION SHOULD CAUSE
2349 ; ;AN ABORT 1ST, THEN A STACK
2350 ; ;VIOLATION
2351 010156 104010 HLT +10 ;IDONTY ABORT OR RECOGNIZE THE
2352 ; ;STACK VIOLATION
2353 010160 000421 BR YELLOW ;GO TO RESET STACK AND RESTORE
2354 ; ;TIMEOUT VECTORS
2355 010162 104011 V1: HLT +11 ;ABORTED BUT STACK VIOLATION
2356 ; ;NOT RECOGNIZED
2357 010164 000417 BR YELLOW ;GO TO RESET STACK AND RESTORE
2358 ; ;TIMEOUT VECTORS
    
```

```

2358 010166 042777 000001 171426 V2: BIC #BIT0,0PARITY ;DISABLE PARITY
2359 010174 005737 000372 TST #372 ;STACK VIOLATION PICKED UP -
2360 ; ;WAS THE PARITY ABORT?
2361 010200 001002 BNE 1$ ;BRANCH IF YES
2362 010202 104012 HLT +12 ;STACK VIOLATION PICKED UP
2363 ; ;BUT ABORT NOT RECOGNIZED
2364 010204 000407 BR YELLOW ;GO TO RESET STACK AND RESTORE
2365 ; ;TIMEOUT VECTORS
2366 010206 012706 001100 1$: MOV #STACK,SP ;RESET STACK BACK TO NORMAL
2367 010212 013746 000372 MOV #372,-(SP) ;PUSH ABORT PC ONTO KERNAL
2368 ; ;STACK OUT OF VIOLATION AREAS
2369 ; ;FOR CHECKING PURPOSES
2370 010216 004037 011550 JSR R0,0CHECKLOC ;CHECK FOR GOOD ABORT
2371 010222 104003 HLT +3 ;ABORTED INCORRECTLY
2372 010224 012706 001100 YELLOW: MOV #STACK,SP ;RESET STACK BACK TO NORMAL
2373 010230 013737 010126 000004 MOV #SAVLOC4,0#4 ;RESTORE CONTENTS OF LOC, 4
2374 010236 005037 000006 CLR #6 ;RESTORE CONTENTS OF LOC, 6
2375 010242 005037 000372 CLR #372 ;RESET TRAPCATCHER LOCATION
2376 ; ;FOR NEXT TEST
2377 010246 005037 000340 CLR #340 ;CLEAR BOTTOM LIMIT LOCATION
2378 ; ;OF 'YELLOW' ZONE USING NORMAL
2379 ; ;IPARITY
    
```

```

;*****
;
;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 536 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, 1 - PS FROM STACK VIOLATION
;
;NOTE: THE PS AND PC FROM THE STACK VIOLATION ARE IN LOCS, 0 & 2
;BECAUSE THE STACK POINTER (R0) 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.

```

*****
*****
*****
2422
2423
2424
2425
2426
2427 010252 000004
2428
2429
2430
2431 010254 004015
2432 010256 010336
2433 010260 005037 000336
2434
2435 010264 042777 000004 171330
2436 010272 012706 000376
2437 010276 013727 000004
2438
2439 010302 000000
2440
2441 010304 012737 010342 000004
2442
2443
2444 010312 012737 000340 000006
2445 010320 012737 010332 001332
2446
2447
2448 010326 005766 177740
2449
2450
2451
2452
2453 010332 104010
2454
2455 010334 000421
2456
2457 010336 104011
2458
2459 010340 000417
2460
2461 010342 042777 000001 171252
2462 010350 005737 000372

```

```

2463
2464 010354 001002
2465 010356 104012
2466
2467 010360 000407
2468
2469 010362 012706 001100
2470 010366 013746 000372
2471
2472
2473 010372 004037 011550
2474 010376 104003
2475 010400 012706 001100
2476 010404 013737 010302 000004
2477 010412 005037 000006
2478 010416 005037 000372
2479
2480 010422 005037 000336
2481
2482
2483
2484
2485 010426 000004
2486
2487
2488
2489 010430 005037 001624
2490
2491 010434 004015
2492 010436 010514
2493 010440 042777 000004 171154
2494 010444 010100
2495 010450 012720 010070
2496
2497
2498 010454 052777 000004 171140
2499 010462 012720 177776
2500
2501 010466 042777 000004 171126
2502 010474 010037 001332
2503
2504
2505 010500 012710 000203
2506
2507
2508 010504 004360 177774
2509 010510 104001
2510 010512 000410
2511 010514 042777 000001 171100
2512 010522 004037 011550
2513 010526 104003
2514 010530 012706 001100
2515
2516

```

```

2517 ;*****
2518 010534 000004 TST611 SCOPZ
2519 ;
2520 ; 11/45 **** ROM STATE 26 ****
2521 ;
2522 ; 11/40 **** ROM STATE 241 ****
2522 010536 004015 JSR R0,(R0) ;SET UP PARITY VECTOR SERVICE
2523 010540 010616 XY ;ROUTINE ADDRESS
2524 010542 042777 000004 171052 BIC #BIT2,0PARITY ;WRITE NORMAL
2525 010550 010100 R1,R0 ;SET UP FOR A DATO
2526 010552 012720 026000 MOV #26000,(R0)+ ;MOVE THE INSTRUCTION
2527 ; ;ICMP #2(R0),R0' TO PARITY
2528 ; ;MEMORY AREA
2529 010556 052777 000004 171036 BIS #BIT2,0PARITY ;WRITE OTHER PARITY
2530 010564 012720 177776 MOV #-2,(R0)+ ;WRITE THE INDEX WCRD IN NEXT
2531 ; ;IPARITY MEMORY AREA LOCATION
2532 010570 042777 000004 171024 BIC #BIT2,0PARITY ;WRITE NORMAL
2533 010576 010037 001332 MOV R0,#SGDDAT ;STORE THE PC THAT SHOULD
2534 ; ;BE PUSHED ON THE STACK
2535 ; ;IF A PARITY ABORT OCCURS
2536 010602 012710 000203 MOV #203,(R0) ;MOVE IRTS R3' INTC NEXT
2537 ; ;IPARITY MEMORY AREA LOCATION
2538 ; ;IN CASE WE DON'T ABORT
2539 010606 004360 177774 JSR R3,-4(R0) ;GO TO PARITY MEMORY AREA
2540 010612 104001 HLT +1 ;IDIDN'T ABORT
2541 010614 000410 BR ,+22 ;GO TO NEXT TEST
2542 010616 042777 000001 170776 YX; BIC #BIT0,0PARITY ;DISABLE PARITY
2543 010624 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT
2544 010630 104003 HLT +3 ;ABORTED INCORRECTLY
2545 010632 012706 001100 MOV #STACK,SP ;RESET THE STACK
2546 ;*****
2547 ;TEST 02 TEST (INDEX WORD) SMO,DM6 MOV INSTRUCTION
2548 ;*****
2549 010636 000004 TST621 SCOPZ
2550 ;
2551 ; 11/45 **** ROM STATE 6 ****
2552 ;
2553 ; 11/40 **** ROM STATE 206 ****
2553 010640 004015 JSR R0,(R0) ;SET UP PARITY VECTOR SERVICE
2554 010642 010720 Y ;ROUTINE ADDRESS
2555 010644 042777 000004 170750 BIC #BIT2,0PARITY ;WRITE NORMAL
2556 010652 010100 R1,R0 ;SET UP FOR A DATO
2557 010654 012720 010060 MOV #10060,(R0)+ ;MOVE THE INSTRUCTION
2558 ; ;ICMP #0,-2(R0)' TO PARITY
2559 ; ;MEMORY AREA
2560 010660 052777 000004 170734 BIS #BIT2,0PARITY ;WRITE OTHER PARITY
2561 010666 012720 177776 MOV #-2,(R0)+ ;WRITE THE INDEX WCRD IN NEXT
2562 ; ;IPARITY MEMORY AREA LOCATION
2563 010672 042777 000004 170722 BIC #BIT2,0PARITY ;WRITE NORMAL
2564 010700 010037 001332 MOV R0,#SGDDAT ;STORE THE PC THAT SHOULD
2565 ; ;BE PUSHED ON THE STACK
2566 ; ;IF A PARITY ABORT OCCURS
2567 010704 012710 000203 MOV #203,(R0) ;MOVE IRTS R3' INTC NEXT
2568 ; ;IPARITY MEMORY AREA LOCATION
2569 ; ;IN CASE WE DON'T ABORT
2570 010710 004360 177774 JSR R3,-4(R0) ;GO TO PARITY MEMORY AREA

```

```

2571 010714 104001 HLT +1 ;IDIDN'T ABORT
2572 010716 000410 BR ,+22 ;GO TO NEXT TEST
2573 010720 042777 000001 170674 Y; BIC #BIT0,0PARITY ;DISABLE PARITY
2574 010726 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT
2575 010732 104003 HLT +3 ;ABORTED INCORRECTLY
2576 010734 012706 001100 MOV #STACK,SP ;RESET THE STACK
2577 ;*****
2578 ;TEST 03 TEST SPL INSTRUCTION (-BOOK)
2579 ;*****
2580 010740 000004 TST631 SCOPZ
2581 ;
2582 ; 11/45 **** ROM STATE 321 ****
2583 ;
2584 ; 11/40 **** ROM STATE 1 ****
2584 010742 004015 JSR R0,(R0) ;SET UP PARITY VECTOR SERVICE
2585 010744 011056 BR ;ROUTINE ADDRESS
2586 010746 042777 000004 170646 BIC #BIT2,0PARITY ;WRITE NORMAL
2587 010754 010102 R1,R2 ;SET UP FOR A DATO
2588 010756 162702 000004 SUB #4,R2 ;CALCULATE THE START
2589 ; ;ADDRESS FOR THIS TEST
2590 010762 012722 012700 MOV #12700,(R2)+ ;MOVE THE INSTRUCTION
2591 ; ;ICMP #-1,R0' TO PARITY
2592 ; ;MEMORY AREA
2593 010766 012722 177777 MOV #-1,(R2)+ ;MOVE -1 INTO NEXT
2594 ; ;IPARITY MEMORY AREA LOCATION
2595 010772 012722 100001 MOV #100001,(R2)+ ;MOVE THE INSTRUCTION
2596 ; ;ICMP #4' INTO NEXT PARITY
2597 ; ;MEMORY AREA LOCATION
2598 010776 052777 000004 170616 BIS #BIT2,0PARITY ;WRITE OTHER PARITY
2599 011004 005737 001642 TST #CPU40 ;ARE WE ON AN 11/40?
2600 011010 001405 BEQ 25 ;BRANCH IF NO
2601 ; ;SINCE THE PC WILL BE UPDATED
2602 ; ;ON THE PARITY ABORT
2603 011012 010237 001332 MOV R2,#SGDDAT ;STORE THIS PC THAT SHOULD
2604 ; ;BE PUSHED ON THE STACK
2605 ; ;IF A PARITY ABORT OCCURS
2606 ; ;SINCE THE PC IS NOT UPDATED
2607 ; ;ON THE PARITY ABORT
2608 011216 012722 000240 MOV #240,(R2)+ ;MOVE A 'NOP' INTO NEXT
2609 ; ;IPARITY MEMORY AREA LOCATION
2610 011022 000404 BR 15 ;CONTINUE WITH TEST
2611 011024 012722 000240 25; MOV #240,(R2)+ ;MOVE A 'NOP' INTO NEXT
2612 ; ;IPARITY MEMORY AREA LOCATION
2613 011030 010237 001332 MOV R2,#SGDDAT ;STORE THE PC THAT SHOULD
2614 ; ;BE PUSHED ON THE STACK
2615 ; ;IF A PARITY ABORT OCCURS
2616 011034 042777 000004 170560 15; BIC #BIT2,0PARITY ;WRITE NORMAL
2617 011042 012712 000203 MOV #203,(R2) ;MOVE IRTS R3' INTC NEXT
2618 ; ;IPARITY MEMORY AREA LOCATION
2619 ; ;IN CASE WE DON'T ABORT
2620 011046 004360 177770 JSR R3,-10(R2) ;GO TO PARITY MEMORY AREA
2621 011052 104001 HLT +1 ;IDIDN'T ABORT
2622 011054 000410 BR ,+22 ;GO TO NEXT TEST
2623 011056 042777 000001 170536 20; BIC #BIT0,0PARITY ;DISABLE PARITY
2624 011064 004037 011550 JSR R0,#CHECKLOC ;CHECK FOR GOOD ABCRT

```

```

2625 011070 104003          HLT      +3          ABORTED INCORRECTLY
2626 011072 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 64          TEST   BNE INSTRUCTION (-BCHK)
;*****
2630 011076 000004          TST64: SCOPE
;
;          11/45 **** ROM STATE 322 ****
;
;
;          11/40 **** ROM STATE 1 ****
2634 011100 004015          JSR      R0,(R0)      ISET UP PARITY VECTOR SERVICE
2635 011102 011214          Z1          IROUTINE ADDRESS
2636 011104 042777 000004 170510  BIC      #BIT2,@PARITY IWRITE NORMAL
2637 011112 010102          MOV      R1,R2        ISET UP FOR A DATO
2638 011114 162702 000004          SUB      #4,R2        ICALCULATE THE START ADDRESS
2639          IFOR THIS TEST
2640 011120 012722 012700    MOV      #12700,(R2)+ IMOVE THE INSTRUCTION
2641          I'MOV #0,R0' INTO PARITY
2642          IMEMORY AREA
2643 011124 012722 000000          MOV      #0,(R2)+     IMOVE A 0 INTO NEXT PARITY
2644          IMEMORY AREA LOCATION
2645 011130 012722 001001          MOV      #1001,(R2)+  IMOVE THE INSTRUCTION
2646          I'BNE ,+4' INTO NEXT
2647          IPARITY MEMORY AREA LOCATION
2648 011134 052777 000004 170460  BIS      #BIT2,@PARITY IWRITE OTHER PARITY
2649 011142 005737 001642          TST      @#CPU40      IARE WE ON AN 11/40?
2650 011146 001405          BEQ      2$          IBRANCH IF NO
2651          ISINCE THE PC WILL BE UPDATED
2652          ION THE PARITY ABORT
2653 011150 010237 001332          MOV      R2,@#SGDDAT  ISTORE THIS PC THAT SHOULD
2654          IBE PUSHED ON THE STACK
2655          IIF A PARITY ABORT OCCURS
2656          ISINCE THE PC WILL NOT BE
2657          IUPDATED ON THE PARITY ABORT
2658 011154 012722 000240          MOV      #240,(R2)+   IMOVE A 'NOP' INTO NEXT
2659          IPARITY MEMORY AREA LOCATION
2660 011160 000404          BR       1$          ICONTINUE WITH TEST
2661 011162 012722 000240 2$1    MOV      #240,(R2)+   IMOVE A 'NOP' INTO NEXT
2662          IPARITY MEMORY AREA LOCATION
2663 011166 010237 001332          MOV      R2,@#SGDDAT  ISTORE THE PC THAT SHOULD
2664          IBE PUSHED ON THE STACK
2665          IIF A PARITY ABORT OCCURS
2666 011172 042777 000004 170422 1$1    BIC      #BIT2,@PARITY IWRITE NORMAL
2667 011200 012712 000203          MOV      #203,(R2)   IMOVE IRTS R3' INTO NEXT
2668          IPARITY MEMORY AREA LOCATION
2669          IIN CASE WE DON'T ABORT
2670 011204 004362 177770          JSR      R3,-10(R2)   IGO TO PARITY MEMORY AREA
2671 011210 104001          HLT      +1          IDIDN'T ABORT
2672 011212 000410          BR       ,+22         IGO TO NEXT TEST
2673 011214 042777 000001 170400 Z11    BIC      #BIT0,@PARITY IDISABLE PARITY
2674 011222 004037 011550          JSR      R0,@#CHECKLOC ICHECK FOR GOOD ABCRT
2675 011226 104003          HLT      +3          IABORTED INCORRECTLY
2676 011230 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 65          TEST   BEQ INSTRUCTION (-BCHK)

```

```

2679          ;*****
2680 011234 000004          TST65: SCOPE
2681          ;
2682          ;          11/45 **** ROM STATE 324 ****
2683          ;
2684          ;          11/40 **** ROM STATE 1 ****
2684 011236 004015          JSR      R0,(R0)      ISET UP PARITY VECTOR SERVICE
2685 011240 011352          Z2          IROUTINE ADDRESS
2686 011242 042777 000004 170352  BIC      #BIT2,@PARITY IWRITE NORMAL
2687 011250 010102          MOV      R1,R2        ISET UP FOR A DATO
2688 011252 162702 000004          SUB      #4,R2        ICALCULATE THE START ADDRESS
2689          IFOR THIS TEST
2690 011256 012722 012700    MOV      #12700,(R2)+ IMOVE THE INSTRUCTION
2691          I'MOV #-1,R0' TO PARITY MEMORY
2692          IAREA
2693 011262 012722 177777          MOV      #-1,(R2)+   IMOVE A -1 INTO NEXT PARITY
2694          IMEMORY AREA LOCATION
2695 011266 012722 001401          MOV      #1401,(R2)+  IMOVE THE INSTRUCTION
2696          I'BEQ ,+4' INTO NEXT PARITY
2697          IMEMORY AREA LOCATION
2698 011272 052777 000004 170322  BIS      #BIT2,@PARITY IWRITE OTHER PARITY
2699 011302 005737 001642          TST      @#CPU40      IARE WE ON AN 11/40?
2700 011304 001405          BEQ      2$          IBRANCH IF NO
2701          ISINCE THE PC WILL BE UPDATED
2702          ION THE PARITY ABORT
2703 011306 010237 001332          MOV      R2,@#SGDDAT  ISTORE THIS PC THAT SHOULD
2704          IBE PUSHED ON THE STACK
2705          IIF A PARITY ABORT OCCURS
2706          ISINCE THE PC WILL NOT BE
2707          IUPDATED ON THE PARITY ABORT
2708 011312 012722 000240          MOV      #240,(R2)+   IMOVE A 'NOP' INTO NEXT
2709          IPARITY MEMORY AREA LOCATION
2710 011316 000404          BR       1$          ICONTINUE WITH TEST
2711 011320 012722 000240 2$1    MOV      #240,(R2)+   IMOVE A 'NOP' INTO NEXT
2712          IPARITY MEMORY AREA LOCATION
2713 011324 010237 001332          MOV      R2,@#SGDDAT  ISTORE THE PC THAT SHOULD
2714          IBE PUSHED ON THE STACK
2715          IIF A PARITY ABORT OCCURS
2716 011330 042777 000004 170264 1$1    BIC      #BIT2,@PARITY IWRITE NORMAL
2717 011336 012712 000203          MOV      #203,(R2)   IMOVE IRTS R3' INTO NEXT
2718          IPARITY MEMORY AREA LOCATION
2719          IIN CASE WE DON'T ABORT
2720 011342 004362 177770          JSR      R3,-10(R2)   IGO TO PARITY MEMORY AREA
2721 011346 104001          HLT      +1          IDIDN'T ABORT
2722 011350 000410          BR       ,+22         IGO TO NEXT TEST
2723 011352 042777 000001 170242 Z21    BIC      #BIT0,@PARITY IDISABLE PARITY
2724 011360 004037 011550          JSR      R0,@#CHECKLOC ICHECK FOR GOOD ABCRT
2725 011364 104003          HLT      +3          IABORTED INCORRECTLY
2726 011366 012706 001100    MOV      #STACK,SP    IRESET THE STACK
;*****
;TEST 66          END OF PROGRAM
;*****
2729 011372 000004          TST66: SCOPE
2730 011374 042777 100005 170220  BIC      #BIT1;BIT2;BIT0,@PARITY IDISABLE ALL PARITY
2731          IAND CLEAR ERROR BIT!
2732

```

```

2733 011402 005737 001630 TST 0#USERTYPE ;DID THE USER SELECT THE REGISTER?
2734 011406 001014 BNE 15 ;BRANCH IF SO AND DON'T STEP
2735 ;UP THE TABLE
2736 011410 002737 000002 001336 ADD #2,0#SREGAD ;STEP UP TO NEXT REGISTER
2737 011416 002737 000002 001414 ADD #2,0#STMPAD ;STEP UP TO CORRESPONDING
2738 ;PARITY MEMORY
2739 011424 002737 000002 001416 ADD #2,0#SSETAD ;STEP UP TO NEXT OFFSET - THIS
2740 ;IS ONLY APPLICABLE IF MEMORY
2741 ;MGMT IS TURNED ON
2742 011432 002737 000002 001446 ADD #2,0#INTERAD ;STEP UP TO NEXT INTERLEAVE
2743 ;VALUE
2744 011440 004337 011474 15: JSR R3,0#FLAGSCLR ;CLEAR PERTINENT FLAGS
2745 011444 000167 000320 JMP SECB ;GO TO RING-A-DING
2746 ;BEFORE REITERATING THE PROGRAM
2747
2748
2749
2750 ;*****
2751 ;
2752 ;ROUTINE FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS
2753 ;
2754 ;*****
2755 011450 012077 170144 VECSETI MOV (R0),0#INTVEC ;WRITE ADDRESS INTO LOCATION 114
2756 011454 052777 000005 170140 BIS #BIT2|BIT0,0#PARITY ;WRITE OTHER PARITY AND DISABLE
2757 011462 000200 RTS R0 ;RETURN TO TESTING
2758 ;*****
2759 ;
2760 ;ROUTINE TO RESET AND GO BACK TO TABLE BEGINNING
2761 ;
2762 ;*****
2763 011464 004337 011506 RESTARTI JSR R3,0#INITIALIZE
2764 011470 000167 000274 JMP SECP ;GO TO RING-A-DING
2765 ;*****
2766 ;
2767 ;ROUTINE TO CLEAR PERTINENT FLAGS BEFORE PASSING THRU THE PROGRAM
2768 ;WITH ANOTHER TABLE ENTRY
2769 ;
2770 ;*****
2771 011474 005037 001624 FLAGSCLR CLR 0#PSPCORZONES ;CLEAR PS, PC AND ZONES ABOPT
2772 CLR 0#MSREGFLAG ;CLEAR MS11 REGISTER PRESENCE
2773 011500 005037 001626 RTS R3 ;FLAG
2774 ;RETURN
2775 011504 000203 ;*****
2776 ;
2777 ;ROUTINE TO COMPLETELY REINITIALIZE BEFORE RESTARTING PROGRAM OVER
2778 ;AT THE BEGINNING OF THE TABLE
2779 ;
2780 ;*****
2781 011506 005037 001624 INITIALIZEI CLR 0#PSPCORZONES ;CLEAR PS, PC AND ZONES
2782 CLR 0#MSREGFLAG ;CLEAR FLAG
2783 011512 005037 001626 CLR 0#MSREGFLAG ;CLEAR MS11 REGISTER PRESENCE
2784 011516 012737 001340 001336 MOV #SREG0,0#SREGAD ;FLAG
2785 ;MOVE #1ST REGISTER CONTAINER
2786

```

```

2787 ;INTO SREGAD WHICH IS USED TO
2788 ;POINT TO THE PARITY REGISTER
2789 ;TABLE
2790 011524 012737 001366 001414 MOV #STMP0,0#STMPAD ;MOVE FIRST MEMORY CONTAINER
2791 ;INTO STMPAD WHICH IS USED TO
2792 ;POINT TO THE MEMORY LOCATION
2793 ;TABLE
2794 011532 012737 001400 001416 MOV #SSET0,0#SSETAD ;MOVE FIRST OFFSET CONTAINER
2795 ;INTO SSETAD WHICH IS USED TO
2796 ;POINT TO THE OFFSET LOCATION
2797 ;TABLE IF MEMORY MGMT IS USED
2798 011540 012737 001450 001446 MOV #INTER0,0#INTERAD ;MOVE 1ST INTERLEAVE CONTAINER
2799 ;INTO INTERAD WHICH IS USED
2800 ;TO POINT TO THE INTERLEAVE
2801 ;LOCATION TABLE
2802 011546 000203 RTS R3 ;RETURN
2803
2804
2805 ;*****
2806 ;
2807 ;SUBROUTINE TO CHECK THAT IF A TEST HAS ABORTED IT DID INDEED
2808 ;ABORT IN THE PROPER PLACE. IT IS QUITE CONCEIVABLE THAT A TEST
2809 ;THAT SHOULD HAVE ABORTED IN THE TWO LOCATION MAP AREA ABORTED
2810 ;ON THE FETCH OF THE INSTRUCTION THAT WAS TO CAUSE THE ABORT,
2811 ;THIS SUBROUTINE WILL FLAG SUCH OCCURRENCES. WITHOUT THIS CHECK
2812 ;THE PROGRAM WOULD APPEAR TO HAVE RUN PROPERLY.
2813 ;
2814 ;BOTH THE CORRECT HIGH ORDER ERROR ADDRESS BITS AND PROPER PC
2815 ;PUSH ON THE STACK ARE LOOKED FOR AFTER A PARITY ABORT OCCURS,
2816 ;
2817 ;*****
2818 011550 010246 CHECKLOC1 MOV R2,-(SP) ;SAVE R2 CONTENTS ON STACK
2819 011552 010346 MOV R3,-(SP) ;SAVE R3 CONTENTS ON STACK
2820 011554 010446 MOV R4,-(SP) ;SAVE R4 CONTENTS ON STACK
2821 011556 005002 CLR R2 ;CLEAR ERROR ADDRESS BIT COMPARE
2822 011560 005004 CLR R4 ;REGISTERS IN CASE WE HAVE AN OLD
2823 ;IMOS DESIGN THAT DOESN'T HAVE
2824 ;ADDRESS BITS
2825 011562 005737 001626 TST 0#MSREGFLAG ;IS AN MS11 OR MS11 PARITY
2826 ;OPTION BEING TESTED?
2827 011566 001027 BNE 75 ;BRANCH IF MS11 AND DON'T DO
2828 ;ADDRESS BITS CHECKING
2829 011570 005737 001624 TST 0#PSPCORZONES ;ARE WE DOING A PS OR PC FETCH
2830 ;FOR ZONE ABORT TEST?
2831 011574 001402 BEQ 55 ;BRANCH IF NO
2832 011576 005002 CLR 02 ;SET THE PARITY REGISTER HIGH
2833 ;ORDER ADDRESS BITS VALUE
2834 ;(BITS 9 THRU 11) TO ZERO FOR
2835 ;THE PS OR PC FETCH ABORT TESTS
2836 011600 000414 BR 25 ;GO CHECK IT AGAINST ACTUAL
2837 ;PARITY REGISTER VALUE
2838 011602 005737 002304 5: TST #MSKT11 ;MEMORY MANAGEMENT ON?
2839 011606 100003 BPL 85 ;BRANCH IF NO
2840 011610 017702 167602 MOV 0#SSETAD,R2 ;PICK UP THE OFFSET VALUE - IT

```

```

2841
2842
2843
2844 011614 000486 017400      BR      2$
2845 011616 012703 017400      MOV     #17400,R3
2846
2847 011622 012702 000140      MOV     #140,R2
2848
2849
2850 011626 020103      1$     CMP     R1,R3
2851
2852 011630 101027      BHI    3$
2853 011632 017704 167764      MOV     @PARITY,R4
2854
2855
2856
2857
2858 011636 042704 170037      BIC    #170037,R4
2859 011642 020402      CMP     R4,R2
2860
2861 011644 001026      BNE    4$
2862 011646 010237 001326      MOV     R2,@#SGOADR
2863
2864 011652 010437 001330      MOV     R4,@#SDOADR
2865
2866 011656 012604      MOV     (SP)+,R4
2867 011660 012603      MOV     (SP)+,R3
2868 011662 012602      MOV     (SP)+,R2
2869
2870
2871
2872
2873
2874 011664 026637 000002 001332      CMP     2(SP),@#SGDDAT
2875
2876 011672 001404      BEQ    9$
2877 011674 016637 000002 001334      MOV     2(SP),@#SDDDAT
2878 011702 000200      RTS    R0
2879 011704 005720      9$     TST    (R2)+
2880
2881 011706 000200      RTS    R0
2882 011710 062703 004000      3$     ADD     #4200,R3
2883
2884 011714 062702 000040      ADD     #40,R2
2885
2886
2887 011720 000742      BR     1$
2888 011722 010237 001326      4$     MOV     R2,@#SGOADR
2889
2890 011726 010437 001330      MOV     R4,@#SDOADR
2891
2892 011732 012604      MOV     (SP)+,R4
2893 011734 012603      MOV     (SP)+,R3
2894 011736 012602      MOV     (SP)+,R2

```

```

;SHOULD BE THE VALLE THAT WILL
;APPEAR IN THE PARITY REGISTER
;ERROR ADDRESS BITS
;GO TO CHECK!
;GET A FIRST POSSIBLE ABORT
;LOCATION AREA
;GET THE FIRST POSSIBLE PARITY
;REGISTER HIGH ORDER ADDRESS
;BITS VALUE (BITS 5 THRU 11)
;IS THIS THE ABORT AREA BEING
;USED?
;NO - SEE IF IT'S THE NEXT ONE
;PROCEED TO SEE IF
;IT WAS A PROPER ABORT BY LOOKING
;AT THE HIGH ORDER ADDRESS BITS
;OF THE PARITY REGISTER
;(BITS 5 THRU 11)
;CLEAR ALL BITS EXCEPT 5 THRU 11
;ARE THE ERROR ADDRESS BITS
;WHAT THEY SHOULD BE?
;BRANCH IF NO
;STORE WHAT THE ADDRESS BITS
;SHOULD HAVE BEEN
;STORE WHAT THE ADDRESS BITS
;WERE
;RESTORE R4 CONTENTS
;RESTORE R3 CONTENTS
;RESTORE R2 CONTENTS

```

```

;IF WE HAVE REACHED THIS PATH 1 OF 2 CONDITIONS EXIST --
;WE HAVE AN OLD MOS DESIGN WITH NO ADDRESS BITS, OR WE HAVE
;CORE OR THE NEW MOS DESIGN WITH ADDRESS BITS OK!!

```

```

2895
2896
2897
2898
2899
2900 011740 026637 000002 001332      CMP     2(SP),@#SGDDAT
2901 011746 001404      BEQ    10$
2902 011750 016637 000002 001334      MOV     2(SP),@#SDDDAT
2903 011756 000200      RTS    R0
2904 011760 016737 167346 001334      10$    MOV     SGDDAT,@#SDDDAT
2905
2906 011766 000200      RTS    R0
2907

```

```

;IF WE HAVE REACHED THIS PATH WE HAVE CORE OR THE NEW MOS DESIGN
;WITH BAD ADDRESS BITS WHICH SHOULD INDICATE AN IMPROPER ABORT!!

```

```

;WAS THE CORRECT PC PUSHED
;ON THE STACK?
;BRANCH IF YES
;SAVE INCORRECT PC FOR PRINTOUT
;GO BACK TO INDICATE BAD ABORT
;RECORD THE FACT THAT THE
;CORRECT PC WAS PUSHED ON STACK
;GO BACK TO INDICATE BAD ABORT

```

```

2908 ;*****
2909 ;END OF PASS ROUTINE
2910 ;INCREMENT THE PASS NUMBER
2911 ;IF SW10=0 DING THE TTY BELL ON END OF PROGRAM
2912 ;IF THERE IS A MONITOR GO TO IT.
2913 ;IF NONE JUMP TO START
2914 011770 000004 SEOP1 SCOPE
2915 011772 005067 167304 CLR STSTNM ;ZERO THE TEST NUMBER
2916 011776 005067 000306 CLR STTIMES ;ZERO THE NUMBER OF ITERATIONS
2917 012002 005267 167272 INC SPASS ;INCREMENT THE PASS NUMBER
2918 012006 032737 002000 177570 BIT #SW10,#SWR ;RING THE BELL?
2919 012014 001002 4S BNE 4S ;NO
2920 012016 104400 012056 TYPE .SBELL ;RING A DING
2921 012022 013700 000042 4SI MOV #42,R0 ;GET MONITOR ADDRESS
2922 012026 001411 BEQ SDOAGN ;IF NONE
2923 012030 022760 177777 000002 CMP #-1,2(R0)
2924 012036 001001 BNE SENDAD
2925 012040 000005 RESET
2926 012042 004710 SENDAD1 JSR PC,(R0) ;GO TO MONITOR
2927 012044 000240 NOP ;SAVE ROOM
2928 012046 000240 NOP ;FOR
2929 012050 000240 NOP ;ACT11
2930 012052 000137 003620 SDOAGN1 JMP #START ;RETURN
2931 012056 177607 000377 SBELL1 .ASCIZ <27><37><37>

```

```

2932 ;*****
2933 ;THIS ROUTINE IS THE SCOPE HANDLER
2934 ;SW14=1 LOOP ON TEST
2935 ;SW11=1 INHIBIT ITERATIONS
2936 ;SW09=1 LOOP ON ERROR
2937 ;SW08=1 LOOP ON TEST IN SW<7;0>
2938 ;THE TEST NUMBER (STSTNM) IS UPDATED AND DISPLAYED
2939
2940 012062 SSCOPE1
2941 012062 006137 177570 ROL #SWR ;LOOP ON PRESENT TEST?
2942 012066 100502 BNI SOVER ;YES IF SW14=1
2943 ;#####START OF CODE FOR THE XOR TESTER#####
2944 012070 000416 EXTSTR1 BR 6S ;IF RUNNING ON THE "XOR" TESTER CHANGE
2945 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
2946 012072 013746 000004 MOV #ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
2947 012076 012737 000004 MOV #55,#ERRVEC ;SET FOR TIMEOUT
2948 012104 005737 177060 TST #177060 ;TIME OUT ON XCR?
2949 012110 012637 000004 MOV (SP)+,#ERRVEC ;RESTORE THE ERROR VECTOR
2950 012114 000497 BR $VLAD ;GO TO THE NEXT TEST
2951 012116 022626 5SI CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
2952 012120 012637 000004 MOV (SP)+,#ERRVEC ;RESTORE THE ERROR VECTOR
2953 012124 000417 BR 7S ;LOOP ON THE PRESENT TEST
2954 012126 6SI ;#####END OF CODE FOR THE XOR TESTER#####
2955 012126 032737 000400 177570 BIT #SW08,#SWR ;LOOP ON SPEC. TEST?
2956 012134 001404 BEQ 2S ;BR IF NO
2957 012136 123767 177570 167136 CMPB #SWR,STSTNM ;DON THE RIGHT TEST? SWR<710>
2958 012144 001453 BEQ SOVER ;BR IF YES
2959 012146 105767 167142 2SI TSTB SERFLG ;HAS AN ERROR OCCURRED?
2960 012152 001415 BEQ 3S ;BR IF NO
2961 012154 032737 001000 177570 BIT #SW09,#SWR ;LOOP ON ERROR?
2962 012162 001404 BEQ 4S ;BR IF NO
2963 012164 014767 167120 167114 7SI MOV $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
2964 012172 000440 BR SOVER
2965 012174 105067 167114 4SI CLRB SERFLG ;ZERO THE ERROR FLAG
2966 012200 005067 000104 CLR STTIMES ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
2967 012204 000415 BR 1S ;ESCAPE TO THE NEXT TEST
2968 012206 032737 004000 177570 3SI BIT #SW11,#SWR ;INHIBIT ITERATIONS?
2969 012214 001011 BNE 1S ;BR IF YES
2970 012216 005767 167056 TST SPASS ;IF FIRST PASS OF PROGRAM
2971 012222 001406 BEQ 1S ; INHIBIT ITERATIONS
2972 012224 005267 167054 INC SICT ;INCREMENT ITERATION COUNT
2973 012230 026767 000054 167046 CMP STTIMES,SICNT ;CHECK THE NUMBER OF ITERATIONS MADE
2974 012236 002016 BGE SOVER ;BR IF MORE ITERATION REQUIRED
2975 012240 012767 000001 167036 1SI MOV #1,SICNT ;REINITIALIZE THE ITERATION COUNTER
2976 012246 016767 000040 000034 MOV $MXCNT,STTIMES ;SET NUMBER OF ITERATIONS TO DO
2977 012254 005267 167022 SSVLAD1 INC STSTNM ;COUNT TEST NUMBERS
2978 012260 011667 167022 MOV (SP),$LPADR ;SAVE SCOPE LOOP ADDRESS
2979 012264 011667 167020 MOV (SP),$LPERR ;SAVE ERROR LOOP ADDRESS
2980 012270 005067 000500 CLR SESCAPE ;CLEAR THE ESCAPE FROM ERROR ADDRESS
2981 012274 016737 167002 177570 SOVER1 MOV STSTNM,#DISPLAY ;DISPLAY TEST NUMBER
2982 012302 016716 167000 MOV $LPADR,(SP) ;FUDGE RETURN ADDRESS
2983 012306 000002 RTI ;FIXES PS
2984 012310 000000 STTIMES 0 ;NUMBER OF ITERATIONS TO PERFORM
2985 012312 000000 SMXCNT 0 ;MAX, NUMBER OF ITERATIONS

```



```

2986 ;*****
2987 ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
2988 ;CALLI
2989 ; ACCEPT ,ADDR ;PUT OCTAL NUMBER IN ADDR
2990 ;
2991 ;SACCEPTI
2992 MOV R0,*(SP) ;PUSH R0 ON STACK
2993 MOV R1,*(SP) ;PUSH R1 ON STACK
2994 MOV R2,*(SP) ;PUSH R2 ON STACK
2995 MOV R3,*(SP) ;PUSH R3 ON STACK
2996 MOV 10(SP),R0 ;GET ADDRESS POINTER OF WHERE TO PUT NUMBER
2997 CLR R1 ;CLEAR PARTIAL NUMBER
2998 MOV #6,R2 ;MAX. # OF DIGITS ALLOWED
2999 GETCHR ;GET ONE CHARACTER
3000 MOV (SP)+,R3 ;AND PUT IT IN R3
3001 MOV R3,R3
3002 TYPE .68 ;ECHO THE CHARACTER
3003 CMP #15,R3 ;WAS THIS CHARACTER A "CR"?
3004 BR 5S ;IF YES
3005 CMP #10,R3 ;INSURE THE CHARACTER IS
3006 BGT 4S ;A DIGIT BETWEEN 0 AND 7.
3007 CMP #17,R3
3008 BLT 4S
3009 DEC R2 ;CHECK NUMBER OF CHARACTERS
3010 BR 4S ;IF TO MANY
3011 ASL R1 ;POSITION PARTIAL NUMBER
3012 ASL R1 ;FOR THIS DIGIT
3013 ASL R1
3014 BIC #C'K7',R3 ;GET RID OF THE ASCII JUNK
3015 BIS R3,R1 ;COMBINE THIS BIT WITH PARTIAL
3016 BR 2S ;GO GET ANOTHER DIGIT
3017 TYPE .SQUES ;TYPE "q"
3018 BR 1S ;GO START OVER
3019 TYPE .SLF ;FOLLOW "CR" WITH A "LF"
3020 MOV R1,(R0)+ ;PASS THE NUMBER TO THE USER
3021 MOV R0,10(SP) ;SET FOR RETURN
3022 MOV (SP)+,R3 ;POPP STACK INTO R3
3023 MOV (SP)+,R2 ;POPP STACK INTO R2
3024 MOV (SP)+,R1 ;POPP STACK INTO R1
3025 MOV (SP)+,R0 ;POPP STACK INTO R0
3026 RTI
3027 .BYTE 0,2 ;STORAGE FOR ASCII CHAR, AND TERMINATOR
3028 .ASCII "? " ;QUESTION MARK
3029 .ASCII "<1>" ;CARRIAGE RETURN
3030 .ASCII "<2>" ;LINEFEED
3031 ;SELECTED REGISTER FROM THE TTY
3032 ;*****
3033 ;THIS ROUTINE INPUTS A SINGLE CHARACTER FROM THE TTY
3034 ;CALLI
3035 ;
3036 ; GETCHR ;INPUT A SINGLE CHARACTER FROM THE TTY
3037 ; RETURN HERE ;CHARACTER IS ON THE STACK
3038 ;
3039 SREADCI MOV (SP),-(SP) ;PUSH DOWN THE PC

```

```

3040 MOV 4(SP),2(SP) ;SAVE THE PS
3041 TSTB #TKS ;WAIT FOR
3042 BPL 1S ;A CHARACTER
3043 MOV #TKB,4(SP) ;READ THE TTY
3044 BIC #C'K177',4(SP) ;GET RID OF JUNK IF ANY
3045 RTI ;GO BACK TO USER
3046 ;*****
3047 ;THIS ROUTINE INPUTS A STRING FROM THE TTY
3048 ;CALLI
3049 ;
3050 ; GETSTR ;INPUT A STRING FROM THE TTY
3051 ; RETURN HERE ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3052 ;TERMINATOR WILL BE A BINARY 0
3053 ;
3054 SREADSI MOV R3,*(SP) ;SAVE R3
3055 MOV #TTYIN,R3 ;GET ADDRESS
3056 CMP #TTYIN+8,,R3 ;BUFFER FULL?
3057 BLOS 4S ;IF YES
3058 GETCHR ;GO READ ONE CHARACTER FROM THE TTY
3059 MOV (SP)+,(R3) ;GET CHARACTER
3060 CMPB #17,(R3) ;IS IT A RUBOUT
3061 BNE 3S ;SKIP IF NOT
3062 TYPE .SQUES ;TYPE A '?'
3063 BR 1S ;ZAP THE BUFFER AND LOOP
3064 MOV (R3),R0 ;ECHO THE CHARACTER
3065 TYPE .68
3066 CMPB #15,(R3)+ ;CHECK FOR RETURN
3067 BNE 2S ;LOOP IF NOT RETURN
3068 CLRB -1,(R3) ;ZAP RETURN (THE 15)
3069 TYPE .SLF ;TYPE A LINE FEED
3070 MOV (SP)+,R3 ;RESTORE R3
3071 MOV (SP),-(SP) ;ADJUST THE STACK AND PUT ADDRESS OF THE
3072 MOV 4(SP),2(SP) ;FIRST ASCII CHARACTER ON IT
3073 MOV #TTYIN,4(SP)
3074 RTI
3075 .BYTE 0 ;RETURN
3076 .BYTE 0 ;TERMINATOR
3077 TKS: 177560 ;TTY KBD STATUS
3078 TKB: 177562 ;TTY KBD BUFFER
3079 STTYINI .BLKB 8. ;RESERVE 8 BYTES FOR TTY INPUT
;FROM THE TTY

```

```

3080
3081
3082
3083
3084
3085
3086
3087
3088 012634
3089 012634 042777 000001 166760
3090 012642 112767 000001 166444
3091 012658 032737 002000 177570
3092 012656 001402
3093 012660 104400 012056
3094 012664 005267 166422 15I
3095 012670 001775
3096 012672 011667 166426
3097 012676 162767 000002 166420
3098 012704 117767 166414 166410
3099 012712 032737 020000 177570
3100 012720 001004
3101 012722 004737 012776
3102 012726 104400 012453
3103 012732 005737 177570 25I
3104 012736 100001
3105 012740 000000
3106 012742 032737 001000 177570 35I
3107 012750 001403
3108 012752 016716 166332
3109 012756 000002
3110 012760 005767 000010 45I
3111 012764 001402
3112 012766 016716 000002
3113 012772 000002 55I
3114 012774 000000
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124 012776 104400 012453
3125
3126 013002 010046
3127 013004 005000
3128 013006 116700 166310
3129 013012 005300
3130 013014 006300
3131 013016 006300
3132 013020 006300
3133 013022 002700 001500

;*****
;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
SHLT:
      BIC    #BIT0,#PARITY
      MOVB  #,$SERFLG
      BIT   #SW10,#SWR
      BEQ  $
      TYPE  ,SBELL
      INC  ,SERTTL
      BEQ  $
      MOV  (SP),SHLTAD
      SUB  #2,SHLTAD
      MOVB #SHLTAD,SITE#B
      BIT  #SW13,#SWR
      BNE  $
      JSR  PC,#TYPERR
      TYPE ,SCTRL
      TST  #SWR
      BPL  $
      HALT
      BIT  #SW09,#SWR
      BEQ  $
      MOV  SLPERR,(SP)
      RTI
      TST  #ESCAPE
      BEQ  $
      MOV  #ESCAPE,(SP)
      RTI
SESCAPE:
      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
;
;*****
TYPERR: TYPE ,SCTRL
;TYPE A CARRIAGE RETURN
;AND LINE FEED
;SAVE R0 CONTENTS
;CLEAR R0
;PICK UP THE ITEM INDEX
;ADJUST THE INDEX
;SO IT WILL WORK FOR
;FOR THE ERROR TABLE
;FORM THE TABLE POINTER

      MOV  R0,-(SP)
      CLR  R0
      MOVB SITE#B,R0
      DEC  R0
      ASL  R0
      ASL  R0
      ASL  R0
      ADD  #SERRTB,R0

```

```

3134 013026 012067 000002
3135 013032 104400
3136 013034 000000
3137 013036 104400 012453 15I
3138
3139 013042 012067 000004
3140 013046 001402
3141 013050 104400
3142 013052 000000 25I
3143 013054 104400 012453 35I
3144
3145 013060 012000
3146 013062 001004
3147 013064 012600 45I
3148 013066 104400 012453
3149
3150 013072 000207
3151 013074
3152 013074 013046 55I
3153
3154 013076 004067 000246
3155 013102 006
3156 013103 001
3157 013104 005710
3158 013106 001766
3159 013110 104400 013116
3160 013114 000767
3161 013116 020040 000040 65I
3162

      MOV  (R0)+,15
      TYPE
      0
      TYPE ,SCTRL
      MOV  (R0)+,25
      BEQ  $
      TYPE
      0
      TYPE ,SCTRL
      MOV  (R0)+,R0
      BNE  $
      MOV  (SP)+,R0
      TYPE ,SCTRL
      RTS  PC
      MOV  0,(R0)+,(SP)
      JSR  R0,$B2OCT
      ,BYTE 6
      ,BYTE 1
      TST  (R0)
      BEQ  $
      TYPE ,6$
      BR  $
      ,ASCIZ / /
      ,EVEN
;PICK UP 'ERROR MESSAGE' POINTER
;TYPE 'ERROR MESSAGE'
;'ERROR MESSAGE' POINTER GOES HERE
;TYPE A CARRIAGE RETURN AND
;LINE FEED
;PICK UP 'DATA HEADER' POINTER
;IF '0' DON'T TYPE
;TYPE 'DATA HEADER'
;'DATA HEADER' POINTER GOES HERE
;TYPE A CARRIAGE RETURN AND
;LINE FEED
;PICK UP 'DATA POINTER'
;IF THERE IS DATA TO TYPE GO DO IT
;RESTORE R0
;TYPE A CARRIAGE RETURN AND
;AND LINE FEED
;RETURN TO TESTING
;SAVE 0(R0)+ FOR TYPEOUT
;TYPE DATA
;GO TYPE--OCTAL ASCII
;TYPE 6 DIGITS
;TYPE LEADING ZEROS
;HAVE WE REACHED THE '0' TERMINATOR
;YES - CLEAN UP FOR RETURN
;TYPE 6 SPACES
;LOOP TILL '0' TERMINATOR REACHED

```

```

3163 ;*****
3164 ;
3165 ;THE FOLLOWING ROUTINE WILL SIZE MEMORY
3166 ;
3167 ;SLSTBLK WILL CONTAIN THE LAST BANK AS AN SAF (SEGMENT ADDRESS FIELD)
3168 ;
3169 ;THE SEGMENT ADDRESS FIELD CONTAINS THE 4 MOST SIGNIFICANT BITS OF
3170 ;THE LAST ADDRESS OF THE LAST BANK FOUND
3171 ;
3172 ;*****
3173 013124 010546 S$SIZE MOV R5,-(SP) ;SAVE R5 CONTENTS
3174 013126 010605 MOV SP,R5 ;SAVE THE STACK POINTER
3175 013130 012703 077406 MOV #77406,R3
3176 013134 010377 169050 MOV R3,#KPPDR0 ;SET THE FOLLOWING PAGE
3177 013140 010377 169046 MOV R3,#KPPDR1 ;DESCRIPTOR REGISTERS TO
3178 013144 010377 169044 MOV R3,#KPPDR2 ;READ/WRITE AND TRANSFER OF
3179 013150 010377 169042 MOV R3,#KPPDR3 ;14096 (10) WORDS PER SEGMENT
3180 013154 010377 169040 MOV R3,#KPPDR4
3181 013160 010377 169036 MOV R3,#KPPDR5
3182 013164 010377 169034 MOV R3,#KPPDR6
3183 013170 010377 169032 MOV R3,#KPPDR7
3184 013174 005077 169030 CLR #KPAR0 ;SET THE FOLLOWING PAGE
3185 013200 012777 000200 169024 MOV #200,#KPAR1 ;ADDRESS REGISTERS TO THEIR
3186 013206 012777 000400 169020 MOV #400,#KPAR2 ;RESPECTIVE OFFSET VALUES
3187 013214 012777 000600 169014 MOV #600,#KPAR3 ;FOR RELOCATION PURPOSES
3188 013222 012777 001000 169010 MOV #1000,#KPAR4
3189 013230 012777 001200 169004 MOV #1200,#KPAR5
3190 013236 012777 001400 169000 MOV #1400,#KPAR6
3191 013244 012777 007600 164774 MOV #7600,#KPAR7
3192 ;THIS ONE'S THE I/C RECORD
3193 ;PAGE CONTAINING CONTROL STATUS
3194 ;REGISTERS, ETC.
3195 013252 016704 164766 MOV KPAR6,R4 ;GET ADDRESS OF PAGE 6 REGISTER
3196 013256 005014 CLR R4 ;CLEAR THE REGISTER
3197 013260 005277 164720 INC #SR0 ;TURN ON MEMORY MANAGEMENT
3198 013264 010746 MOV PC,-(SP) ;MAKE K11 TIMEOUT SERVICE
3199 013266 062716 ADD #SKTOUT-,(SP) ;ROUTINE ADDRESS POSITION
3200 ;INDEPENDENT
3201 013272 012637 000004 MOV (SP)+,#ERRVEC ;SET FOR TIMEOUT
3202 013276 005737 143776 1$1 TST #143776 ;TRAP ON NON-EXISTENT MEMORY
3203 013302 062714 000040 ADD #40,(R4) ;MAKE A 1K STEP
3204 013306 027714 164734 CMP #KPAR7,(R4) ;LAST ONE?
3205 013312 003371 BGT 1$ ;NO - TRY IT!
3206 013314 011400 SKTOUT MOV R4,R0 ;GET LAST BANK +1
3207 013316 162700 000040 SUB #40,R0 ;DROP BACK
3208 013322 012737 000006 000004 MOV #6,#ERRVEC ;SET FOR ERRORS
3209 013330 010506 MOV R5,SP ;RESTORE THE STACK POINTER
3210 013336 010067 000010 MOV R0,SLSTBLK ;STORE THE SAF
3211 013342 012605 CLR #SR0 ;TURN MEMORY MGMT OFF
3212 013344 000207 1$1 MOV (SP)+,R5 ;RESTORE R5
3213 013346 000000 PC RTS ;RETURN TO NORMAL FLOW
3214 ;CONTAINS THE SAF
3215 ;*****
3216 ;BINARY TO OCTAL (ASCII) AND TYPE
;SB2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE

```

```

3217 ;CALLI
3218 ; MOV NUM,-(SP) ;NUMBER TO BE TYPED
3219 ; JSR R0,$B2OCT ;CALL FOR TYPEOUT
3220 ; ;BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3221 ; ;BYTE M ;M=1 OR 0
3222 ; ; ;I=TYPE LEADING ZEROS
3223 ; ; ;J=SUPPRESS LEADING ZEROS
3224 ;
3225 ;SB20i---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST SB2OCT OR SB2016
3226 ;CALLI
3227 ; MOV NUM,-(SP)
3228 ; JSR R0,$B201
3229 ;
3230 ;SB2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3231 ;CALLI
3232 ; MOV NUM,-(SP)
3233 ; JSR R0,$B2016
3234 ;
3235 013350 112067 000201 SB20CT MOVB (R0)+,$OMODE*1 ;PICKUP# THE NUMBER OF DIGITS TO TYPE
3236 013354 112067 000173 MOVB #0,$OFILL ;GET THE ZERO FILL SWITCH
3237 013360 000406 BR $B201
3238 013362 112767 000001 000163 SB2016 MOVB #1,$OFILL ;SET THE ZERO FILL SWITCH
3239 013370 112767 000006 000157 MOVB #6,$OMODE*1 ;SET FOR SIX(6) DIGITS
3240 013376 112767 000005 000146 SB201i MOVB #5,$OCNT ;SET THE ITERATION COUNT
3241 013404 010346 MOV R3,-(SP) ;SAVE R3
3242 013406 010446 MOV R4,-(SP) ;SAVE R4
3243 013410 010546 MOV R5,-(SP) ;SAVE R5
3244 013412 116784 000137 MOVB $OMODE*1,R4 ;GET THE NUMBER OF DIGITS TO TYPE
3245 013416 005404 NEG R4
3246 013420 062784 000006 #6,R4 ;SUBTRACT IT FOR MAX. ALLOWED
3247 013424 110467 000124 MOVB R4,$OMODE ;SAVE IT FOR USE
3248 013430 116704 000117 MOVB $OFILL,R4 ;GET THE ZERO FILL SWITCH
3249 013434 016605 000010 MOV #0,(SP),R5 ;PICKUP# THE INPUT NUMBER
3250 013440 005003 CLR R3 ;CLEAR THE OUTPUT WORD
3251 013442 006105 1$1 ROL R5 ;ROTATE MSB INTO "C"
3252 013444 000404 BR 3$ ;GO DO MSB
3253 013446 006105 2$1 ROL R5 ;FORM THIS DIGIT
3254 013450 006105 ROL R5
3255 013452 026105 ROL R5
3256 013454 010503 MOV R5,R3
3257 013456 006173 3$1 ROL R3 ;GET LSB OF THIS DIGIT
3258 013460 105367 000070 DECB $OMODE ;TYPE THIS DIGIT?
3259 013464 100016 BR 7$ ;BR IF NO
3260 013466 042703 177770 BIC #177770,R3 ;GET RID OF JUNK
3261 013472 001002 BNE 4$ ;TEST FOR 0
3262 013474 005704 TST R4 ;SUPPRESS THIS 0?
3263 013476 001403 BR 8$ ;BR IF YES
3264 013500 005204 4$1 INC R4 ;DON'T SUPPRESS ANYMORE 0'S
3265 013502 052703 000060 BIS #10,R3 ;MAKE THIS DIGIT ASCII
3266 013506 052703 000040 BIS #1,R3 ;MAKE ASCII IF NOT ALREADY
3267 013512 110367 000032 MOVB R3,R0 ;SAVE FOR TYPING
3268 013516 104400 013550 TYPE #8 ;GO TYPE THIS DIGIT
3269 013522 105367 000024 7$1 DECB $OCNT ;COUNT BY 1
3270 013526 003347 BGT 2$ ;BR IF MORE TO DO

```

3271	013530	002402	BLT	6S	IBR IF DONE
3272	013532	005204	INC	R4	IINSURE LAST DIGIT ISN'T A BLANK
3273	013534	000744	BR	2S	IGO DO THE LAST DIGIT
3274	013536	012605	6S; MOV	(SP)+,R5	IRESTORE R5
3275	013540	012604	MOV	(SP)+,R4	IRESTORE R4
3276	013542	012603	MOV	(SP)+,R3	IRESTORE R3
3277	013544	012616	MOV	(SP)+,(SP)	ISET THE STACK FOR RETURNING
3278	013546	000200	RTS	R0	IRETURN
3279	013550	000	6S; ,BYTE	0	I STORAGE FOR ASCII DIGIT
3280	013551	000	,BYTE	0	ITERMINATOR FOR TYPE ROUTINE
3281	013552	000	SOCNT; ,BYTE	0	IOCTAL DIGIT COUNTER
3282	013553	000	SOFILL; ,BYTE	0	IZERO FILL SWITCH
3283	013554	000000	SOMODE; 0		INUMBER OF DIGITS TO TYPE

3284					
3285					
3286					
3287	013556	010046			
3288	013560	010000	000002		
3289	013564	005740			
3290	013566	111000			
3291	013570	010000	013576		
3292	013574	000200			
3293					
3294					
3295					
3296					
3297					
3298	013576	001110			
3299	013600	012456			
3300	013602	012512			
3301	013604	012314			

```

;*****
;TRAP HANDLER
STRAP;  MOV  R0,-(SP)      ;SAVE R0
        MOV  2(SP),R0     ;GET TRAP ADDRESS
        TST  -(R0)        ;BACKUP BY 2
        MOVB (R0),R0      ;GET RIGHT BYTE OF TRAP
        MOV  STRPAD(R0),R0 ;INDEX TO TABLE
        RTS  R0           ;GO TO ROUTINE

;TRAP TABLE
; ROUTINE
; -----
STRPAD;  STYPE  ICALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
        SREADC ;CALL=GETCHR   TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
        SREADS ICALL=GETSTR   TRAP+4(104404)  TTY TYPEIN STRING ROUTINE
        SACCEPT ICALL=ACCEPT TRAP+6(104406)  READ AN OCTAL NUMBER FROM TTY
    
```

```

3302
3303
3304
3305 013606 012737 013734 000024
3306 013614 012737 000340 000026
3307 013622 010046
3308 013624 010146
3309 013626 010246
3310 013630 010346
3311 013632 010446
3312 013634 010546
3313 013636 010667 000076
3314 013642 012737 013654 000024
3315 013650 000000
3316 013652 000776
3317
3318
3319 013654 016706 000060
3320 013660 005067 000054
3321 013664 005267 000050
3322 013670 001375
3323 013672 012605
3324 013674 012604
3325 013676 012603
3326 013700 012602
3327 013702 012601
3328 013704 012600
3329 013706 012737 013606 000024
3330 013714 012737 000340 000026
3331 013722 004400 013742
3332 013726 012716 001706
3333 013732 000002
3334 013734 000000
3335 013736 000776
3336 013740 000000
3337 013742 005015 047520 042527
3338 013750 000122
3339

```

```

;*****
;POWER DOWN ROUTINE
SPWRDN: MOV    $SILLUP,$#PWRVEC      ISET FOR FAST UP
        MOV    R0,-(SP)             IPRIO17
        MOV    R1,-(SP)             IPUSH R0 ON STACK
        MOV    R2,-(SP)             IPUSH R1 ON STACK
        MOV    R3,-(SP)             IPUSH R2 ON STACK
        MOV    R4,-(SP)             IPUSH R3 ON STACK
        MOV    R5,-(SP)             IPUSH R4 ON STACK
        MOV    SP,$SAVR6           IPUSH R5 ON STACK
        MOV    $SPWRUP,$#PWRVEC     ISAVE SP
        HALT                          ISET UR VECTOR
        BR     .-2                    IHANG UP
;*****
;POWER UP ROUTINE
SPWRUP: MOV    $SAVR6,SP           IGET SP
        CLR    $SAVR6              IWAIT LOOP FOR THE TTY
        INC    $SAVR6              IWAIT FOR THE INC
        BNE    13                    IOP WORD
        MOV    (SP)+,R5             IPOP STACK INTO R5
        MOV    (SP)+,R4             IPOP STACK INTO R4
        MOV    (SP)+,R3             IPOP STACK INTO R3
        MOV    (SP)+,R2             IPOP STACK INTO R2
        MOV    (SP)+,R1             IPOP STACK INTO R1
        MOV    (SP)+,R0             IPOP STACK INTO R0
        MOV    $SPWRDN,$#PWRVEC     ISET UP THE POWER DOWN VECTOR
        MOV    R0,$#PWRVEC+2        IPRIO17
        TYPE   $POWER               IPOWER FAIL MESSAGE
        MOV    $BEGIN,(SP)          IRESTART AT BEGIN
        RTI
        HALT                          ITHE POWER UP SEQUENCE WAS STARTED
        BR     .-2                    BEFORE THE POWER DOWN WAS COMPLETE
        $SAVR6 0
        SPOWER1 ,ASCIZ <1><12>"POWER"
        INPUT THE SP HERE
;*****
;EVEN

```

```

3340
3341
3342
3343
3344
3345
3346 013752 042524 052123 042040 EM1: .ASCIZ /TEST DIDN'T ABORT /
3347 013760 042111 023516 020124
3348 013766 041101 051117 020124
3349 013774 020040
3350 013776 040506 040524 020114 EM2: .ASCIZ /FATAL ERROR TO PROGRAM /
3351 014004 051105 047522 020122
3352 014012 047524 050040 047522
3353 014020 051107 046501 020040
3354 014026 000
3355 014027 101 047502 052122 EM3: .ASCIZ /ABORTED INCORRECTLY /
3356 014034 042105 044400 041516
3357 014042 051117 042522 052103
3358 014050 054514 020040 000
3359 014055 116 020117 040520 EM4: .ASCIZ /NO PARITY MEMORY FOUND BELOW 28K /
3360 014062 044522 054524 046440
3361 014070 046505 051117 020131
3362 014076 047506 047125 020104
3363 014104 042502 047514 020127
3364 014112 034062 020113 000040
3365 014120 042522 042523 020124 EM5: .ASCIZ /RESET DOESN'T WORK /
3366 014126 047504 051505 023516
3367 014134 020124 047527 045522
3368 014142 020040 000
3369 014145 125 042523 020122 EM6: .ASCIZ /USER SELECTED REGISTER NOT PRESENT /
3370 014152 042523 042514 052103
3371 014160 042105 051040 043505
3372 014166 051511 042524 020122
3373 014174 047516 020124 051120
3374 014202 051505 047105 020124
3375 014210 000040
3376 014212 047516 050040 051101 EM7: .ASCIZ /NO PARITY MEMORY FOUND AT ALL /
3377 014220 052111 020131 042515
3378 014226 047515 054522 043040
3379 014234 052517 042116 040440
3380 014242 020124 046101 020114
3381 014250 000040
3382 014252 044504 047104 052047 EM10: .ASCIZ /DIDN'T ABORT OR RECOGNIZE STACK VIOLATION /
3383 014260 040440 047502 052122
3384 014266 047440 020122 042522
3385 014274 047503 047107 055111
3386 014302 020105 052123 041501
3387 014310 020113 044526 046117
3388 014316 052101 047511 020116
3389 014324 000040
3390 014326 041101 051117 042524 EM11: .ASCIZ /ABORTED BUT STACK VIOLATION NOT RECOGNIZED /
3391 014334 020104 052502 020124
3392 014342 052123 041501 020113
3393 014350 044526 046117 052101

```

3394	014356	047511	020116	047516									
3395	014364	020124	042522	047503									
3396	014372	047107	050111	042105									
3397	014400	020040	000										
3398	014403	123	040524	045503	EM12:	.ASCII	/STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED	/					
3399	014410	053040	047511	040514									
3400	014416	044924	047117	050040									
3401	014424	041911	042513	020104									
3402	014432	050125	041040	052125									
3403	014440	040440	047502	052122									
3404	014446	047040	052117	051040									
3405	014454	041505	043517	044516									
3406	014462	042932	020104	000040									
3407	014470	051120	043517	040522	DH1:	.ASCII	/PROGRAM REGISTER	<15><12>					
3408	014476	020115	020040	042522									
3409	014504	044507	052123	051105									
3410	014512	005015											
3411	014514	020040	041520	020040		.ASCII	/ PC	UNDER TEST/					
3412	014522	020040	052440	042116									
3413	014530	051105	052040	051505									
3414	014536	000124											
3415	014540	051040	043505	051511	DH2:	.ASCII	/ REGISTER	<15><12>					
3416	014546	042524	006522	012									
3417	014553	125	042116	051105		.ASCII	/ UNDER TEST/						
3418	014560	052040	051505	020124									
3419	014566	051120	043517	040522	DH3:	.ASCII	/ PROGRAM	<15><12>					
3420	014574	006515	012										
3421	014577	040	050040	000103		.ASCII	/ PC/						
3422	014604	051120	043517	040522	DH4:	.ASCII	/ PROGRAM REGISTER	EXPECTED	ACTUAL	EXPECTED	ACTUAL	<15><12>	
3423	014612	020115	020040	042522									
3424	014620	044507	052123	051105									
3425	014626	020040	020040	054105									
3426	014634	042520	052103	042105									
3427	014642	020040	020040	040440									
3428	014650	052103	040525	020114									
3429	014656	020040	042440	050130									
3430	014664	041505	042524	020104									
3431	014672	020040	041501	052524									
3432	014700	046101	005015										
3433	014704	020040	041520	020040		.ASCII	/ PC	UNDER TEST	ADDR. BITS	ADDR. BITS	ABORT PC	ABORT PC/	
3434	014712	020040	052440	042116									
3435	014720	051105	052040	051505									
3436	014726	020124	040440	042104									
3437	014734	027122	041040	052111									
3438	014742	020123	040440	042104									
3439	014750	027122	041040	052111									
3440	014756	020123	040440	047502									
3441	014764	052122	050040	020103									
3442	014772	020040	041101	051117									
3443	015000	020124	041520	000									
3444	015005	120	047522	051107	DH5:	.ASCII	/ PROGRAM REGISTER	EXPECTED	<15><12>				
3445	015012	046501	020040	051040									
3446	015020	043505	051511	042524									
3447	015026	020122	020040	042440									

3448	015034	050130	041505	042524									
3449	015042	006504	012										
3450	015045	040	050040	020103		.ASCII	/ PC	UNDER TEST	ABORT #C/				
3451	015052	020040	020040	052440									
3452	015060	042116	051105	052040									
3453	015066	051505	020124	020040									
3454	015074	041101	051117	020124									
3455	015102	041520	000										
3456		015106				.EVEN							
3457	015106	001324	001622	000000	DT1:	.WORD	SMLTD,PARITY,0						
3458	015114	001622	000000		DT2:	.WORD	PARITY,0						
3459	015120	001324	000000		DT3:	.WORD	SMLTD,0						
3460	015124	001324	001622	001326	DT4:	.WORD	SMLTD,PARITY,\$G0ADR,\$B0ADR,\$G0AT,\$B0CAT,0						
3461	015132	001330	001332	001334									
3462	015140	000000											
3463	015142	001324	001622	001332	DT5:	.WORD	SMLTD,PARITY,\$G0DAT,0						
3464	015150	000000											
3465													
3466		000001				.END							

CROSS REFERENCE TABLE
ADC 200 630 633 636 650 656 682 688 715 728 737 871 891 950 1233
DCKBR.P11 CROSS REFERENCE TABLE
ASL 544 3011 3012 3013 3130 3131 3132
BEC 191 588 606 649 660 676 681 692 712 725 733 785 797 810 823
RGE 2974
BGT 3006 3204 3270
BIC 783 795 811 859 865 989 992 1011 1014 1033 1036 1055 1058 1077 1080
BIS 775 791 807 819 1163 1209 1302 1326 1401 1618 1723 1780 1823 1886 2008
BIT 586 776 784 792 796 809 822 2018 2955 2961 2968 3091 3099 3106
BLCS 3056
BLT 208 3008 3010 3271
BME 548 804 2942
BNE 197 204 777 793 1633 2361 2464 2734 2827 2861 2919 2924 2969 3060 3066
BPL 210 612 629 644 958 2039 3042 3104 3259
RR 193 210 531 555 591 599 638 651 665 673 683 697 705 717 949
CLR 483 490 491 492 494 499 497 500 501 502 503 504 505 506 507
CMFB 533 556 611 675 718 731 758 869 873 1076 1098 1120 1142 1167 1189
DEC 549 645 677 1609 1615 1621 1705 1729 1734 1818 3009 3129
DEC 207 3258 3269
HALT 127 192 3105 3315 3334
INC 534 554 597 826 868 2158 2917 2972 2977 3094 3196 3264 3272 3321
IOT 35
JMP 130 613 713 726 730 736 739 744 882 896 950 1488 1510 1532 2157

CROSS REFERENCE TABLE
JSR 2745 2764 2930 202 209 552 561 615 640 843 854 965 971 981 993 1003 1015 1025
MOV 194 195 199 205 480 481 482 484 485 486 487 488 489 526 527
NEG 3245
NOP 2927 2928 2929
RESET 808 2925
ROL 2941 3251 3253 3254 3255 3257
RTI 201 1948 2009 2983 3026 3049 3073 3109 3113 3333
RTS 214 761 825 829 862 872 1646 1745 1891 2757 2775 2802 2878 2881 2903
SOB 1977
SPL 530
SUB 754 952 1713 1836 2097 2588 2630 2688 3097 3206
TRAP 3294 3299 3300 3301
TST 198 542 605 614 628 638 643 647 659 679 691 711 724 732 803

DCKBR.C.P11 CROSS REFERENCE TABLE

Table with 13 columns representing numerical values and 15 rows of categorical labels such as TSTB, .ASCIZ, .BLKB, .BYTE, .ENABL, .EAND, .EADC, .EQUIV, .EVEN, .IF, .IFT, .IFTF, .IIF, .IRP, .LIST, .MACRO, .MCALL, .NLIST, .PAGE, .REPT, .SBTTL, .TITLE, and .WCRC.

DCKBR.C.P11 CROSS REFERENCE TABLE

Table with 13 columns representing numerical values and 15 rows of categorical labels such as .MACRO, .MCALL, .NLIST, .PAGE, .REPT, .SBTTL, .TITLE, and .WCRC.

MAINDEC-11-DCKBR-C MACY11.624 14-MAR-74 14:48 PAGE 87
DCKBR.P11

*DCKBR;DCKBR/SOL/CRP*DCKBR.P11/N
RLN-TIME: 33 29 5 SECONDS
CCRE USED: 13K