

11/40/45

CPU PARITY TEST
MD-11-DCKBR-E

EP DCKBR E DL A

NOV 1976

COPYRIGHT 1976

digital

FICHE 1 OF 1

MADE IN USA

Table with multiple columns and rows of data, including headers like 'DCKBRE SEQ' and various numerical values.

Small table or data block in the bottom right corner.

CONTENTS

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
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

1.	ABSTRACT
2.	REQUIREMENTS
3.	EQUIPMENT
4.	STORAGE
5.	PRELIMINARY PROGRAMS
6.	LOADING PROCEDURE
7.	STARTING PROCEDURE
8.	CONTROL SWITCH SETTINGS
9.	STARTING ADDRESS
10.	OPERATOR ACTION
11.	OPERATIONAL SWITCH SETTINGS
12.	SPECIAL USAGE
13.	SPECIAL NOTE ON SW<12>
14.	SUBROUTINE ABSTRACTS
15.	THRU
16.	SUBROUTINES EXPLAINED INDIVIDUALLY
17.	6.15
18.	ERROR PRINTOUTS
19.	7.1
20.	THRU
21.	ERROR PRINTOUT EXAMPLES AND EXPLANATIONS
22.	7.10
23.	RESTRICTIONS
24.	MISCELLANEOUS
25.	9.1
26.	EXECUTION TIME
27.	9.2
28.	PROGRAM TABLE LOCATIONS
29.	9.3
30.	PROGRAM TABLE SETUP WITH KT11 ENABLED
31.	9.4
32.	PROGRAM TABLE SETUP WITH KT11 DISABLED
33.	9.5
34.	STACK POINTER
35.	9.6
36.	MAINTENANCE HINT
37.	10.
38.	PROGRAM DESCRIPTION
39.	10.1
40.	PROGRAM FLOW DIAGRAM

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
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

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.

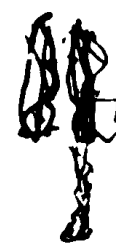
4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE PARAGRAPH 5.

4.2 STARTING ADDRESS

THE PROGRAM IS STARTED AT ADDRESS 200.



2

131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186

4.3 OPERATOR ACTION

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

5. OPERATIONAL SWITCH SETTINGS

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

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

- (1) THE USER WISHES TO SELECT A PARTICULAR REGISTER TO UNDERGO TESTING, NOT USE THE KT11, AND LOOP ON TST37
 - (A) LOAD ADDRESS 200
 - (B) SET SWITCHES 6 AND 12
 - (C) HIT START
 - (D) THE TELETYPE WILL RESPOND BY ASKING THE USER TO 'TYPE THE REGISTER YOU DESIRE & HIT CARRIAGE RETURN' AND WILL WAIT FOR THIS RESPONSE
E.G. 172110 (NOT 772110)
 - (E) BEFORE TYPING A REPLY AND HITTING A CARRIAGE RETURN, PUT SW<06> AND SW<12> DOWN, SET SW<08>, AND PLACE THE VALUE 37 INTO SWS<07 THRU 00>
 - (F) TYPE THE RESPONSE AND HIT CARRIAGE RETURN
 - (G) YOU SHOULD BE LOOPING ON TST37 WHICH CAN BE EASILY VERIFIED BY EXAMINING THE CONTENTS OF \$LPADR

NOTE: LOOPING ON A PARTICULAR TEST CAPABILITY WILL ONLY WORK WHEN THE USER HAS SELECTED A PARTICULAR REGISTER

101

MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 5

187

USING THE SW<12> OPTION

188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236

PAGE 5

(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.

237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290

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 FLAGSCLR

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

6.8 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 TYPERR ROUTINE WHICH TYPES OUT THE ERROR MESSAGES AND DATA HEADERS IS CALLED WITHIN THIS ROUTINE.

6.9 INITIALIZE

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

291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343

6.10 PARTST

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

6.11 \$PWRDN

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

6.12 \$\$SCOPE

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

6.13 TRAPCATCHER

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

6.14 TYPERR

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

6.15 VECSET

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

344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

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

NO1.

MA'DEC-11-DCKBR-E
DCKBR.E.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 10

400
401
402
403
404

(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

~~71~~ 71

R

C02

MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 12

461

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

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

7.8 HLT +10
 DIDN'T ABORT OR RECOGNIZE STACK VIOLATION
 PROGRAM REGISTER EXPECTED
 PC UNDER TEST ABORT PC
 ** APPROPRIATE VALUES **

7.9 HLT +11
 ABORTED BUT STACK VIOLATION NOT RECOGNIZED
 PROGRAM REGISTER
 PC UNDER TEST
 ** APPROPRIATE VALUES **

7.10 HLT +12
 STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED
 PROGRAM REGISTER
 PC UNDER TEST
 ** APPROPRIATE VALUES **

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

9. MISCELLANEOUS

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

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

46
47
48
49
50
51
52
53
54
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
109
110
111
112
113
114
115

516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571

PAGE 11

THE ABSOLUTE CORE LOCATIONS FOR TABLE ENTRYS ARE AS FOLLOWS:

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

LOCATION 1342 UP TO 1364 WILL CONTAIN ANYMORE
REGISTERS FOUND

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

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

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

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

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

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

9.3 PROGRAM TABLE SET UP WITH KT11 ENABLED

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

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

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

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

572
573

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)

574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629

630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685

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 BK 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

686
697

(9) THE MICROSTATE VALUE THAT WAS IN THE CONSOLE DATA REGISTER JUST BEFORE IT TURNED 0 WAS THE ABORT MICROSTATE.

8

688
699
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743

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 SMO,DMO' INSTRUCTION
- (D) THE 'POP' ON AN RTS INSTRUCTION
- (E) 1ST AND 2ND 'POP' ON AN RTI INSTRUCTION
- (F) PS AN PC FETCH INSTRUCTIONS
- (G) INDEXED WORD INSTRUCTIONS (DM6,DM7,SM6)
- (H) CONDITIONAL BRANCH NOT OK INSTRUCTIONS
- (I) STACK VIOLATIONS IN 'RED' AND 'YELLOW' ZONES

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

10.1 PROGRAM FLOW DIAGRAM

%

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

;OPERATIONAL SWITCH SETTINGS

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
11	INHIBIT ITERATIONS
10	0 - BELL ON PASS COMPLETE 1 - BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SW<7:0>
;SPECIAL USER TYPE SWITCH SW<12>	
;IF SET INDICATES USER INPUT	




```

800
801
802
803      100000
804      040000
805      020000
806      010000
807      004000
808      002000
809      001000
810      000400
811      000200
812      000100
813      000040
814      000020
815      000010
816      000004
817      000002
818      000001
819
820
821
822
823
824
825
826
827
828
829
830
831      000004
832      000010
833      000014
834      000014
835      000014
836      000020
837      000024
838      000030
839      000034
840      000001
841
842
843
844
845      000000
846
847
848
849      000200
850
851      000200 000137 001706
852
853      000046
854      000046 012042
855      000052

```

.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

```

:VECTOR ADDRESSES

```

ERRVEC= 4
RESVEC= 10
TBITVEC=14
TRTVEC= 14
BPTVEC= 14
IOTVEC= 20
PWRVEC= 24
EMTVEC= 30
TRAPVEC=34
N=1

```

```

;DEFINE STARTING 'N' FOR SCOPE
;ROUTINE WHICH PRINTS TEST NO.
;AND APPROPRIATE COMMENT

```

```

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

```

. =200

JMP @#BEGIN ;JUMP TO STARTING ADDRESS OF PROGRAM

. =46
\$ENDAD

. =52

856 000052 040000

BIT14

857

858 000204

. =204

859

850

;KT11-D STATUS REGISTER ADDRESSES

861

862 000204 177572

SR0: 177572

863 000206 177576

SR2: 177576

864

865

;KERNAL PAGE DESCRIPTOR REGISTERS

866

867 000210 172300

KPDR0: 172300

868 000212 172302

KPDR1: 172302

869 000214 172304

KPDR2: 172304

870 000216 172306

KPDR3: 172306

871 000220 172310

KPDR4: 172310

872 000222 172312

KPDR5: 172312

873 000224 172314

KPDR6: 172314

874 000226 172316

KPDR7: 172316

875

876

;KERNAL PAGE ADDRESS REGISTERS

877

878 000230 172340

KPAR0: 172340

879 000232 172342

KPAR1: 172342

880 000234 172344

KPAR2: 172344

881 000236 172346

KPAR3: 172346

882 000240 172350

KPAR4: 172350

883 000242 172352

KPAR5: 172352

884 000244 172354

KPAR6: 172354

885

886

887 000246 172356

KPAR7: 172356

888

;KT11 VECTOR ADDRESS

889 000250 000250 000252

SEGVEC: 250,252

```

890
891
892          001100
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911 001100 177564
912 001102 177566
913 001104   000
914 001105   002
915 001106   000
916 001107   000
917
918 001110 105767 177772
919 001114 001402
920 001116 000000
921 001120 000407
922 001122 010046
923 001124 017600 000002
924 001130 112046
925 001132 001005
926 001134 005726
927 001136 012600
928 001140 062716 000002
929 001144 000002
930 001146 004767 000026
931 001152 122726 000012
932 001156 001364
933 001160 016746 177720
934
935 001164 105366 000001
936 001170 002770
937 001172 004767 000002
938 001176 000772
939 001200 105777 177674
940 001204 100375
941 001206 116677 000002 177666
942 001214 000207
943 001216 000062

;*****
;.=1107
;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.

;CALL:
;1) USING A TRAP INSTRUCTION
;   TYPE      ,MESADR          ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;OR
;   TYPE
;   MESADR
;2) USING A JSR INSTRUCTION
;   MOV      PS,-(SP)          ;PUSH PROCESSOR STATUS WORD ON THE STACK
;   JSR      PC,$TYPE         ;CALL TYPE ROUTINE
;   MESADDR          ;FIRST ADDRESS OF MESSAGE

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

$TYPE: TSTB $TPFLG    ;IS THERE A TERMINAL?
        BEQ 6$        ;BR IF YES
        HALT          ;HALT HERE IF NO TERMINAL
        BR 7$        ;LEAVE
6$: MOV RO,-(SP)      ;SAVE RO
        MOV @2(SP),RO ;GET ADDRESS OF ASCIZ STRING
1$: MOVB (RO)+,-(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
        BNE 2$        ;BR IF IT ISN'T THE TERMINATOR
        TST (SP)+     ;IF TERMINATOR POP IT OFF THE STACK
        MOV (SP)+,RO  ;RESTORE RO
7$: ADD #2,(SP)      ;ADJUST RETURN PC
        RTI          ;RETURN
2$: JSR PC,5$        ;GO TYPE THIS CHARACTER
3$: CMPB #12,(SP)+   ;CHECK IF THE CHAR. TYPED WAS A LINE FEED
        BNE 1$        ;GO GET NEXT CHAR. IF NOT LINE FEED
        MOV $NULL,-(SP) ;GET # OF FILLER CHARS. NEEDED
                        ;AND THE NULL CHAR.
4$: DECB 1(SP)       ;DOES A NULL NEED TO BE TYPED?
        BLT 3$        ;BR IF NO--GO POP THE NULL OFF OF STACK
        JSR PC,5$    ;GO TYPE A NULL
        BR 4$        ;LOOP
5$: TSTB @2$TPS     ;WAIT UNTIL PRINTER IS READY
        BPL 5$
        MOVB 2(SP),@2$TPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
        RTS PC
        .BLKB 62     ;RESERVE SOME MORE CORE FOR OVERLAY CAPIBILITIES

```


1000	001436	000000
1001	001440	000000
1002	001442	000000
1003	001444	000000
1004	001446	000000
1005	001450	000000
1006	001452	000000
1007	001454	000000
1008	001456	000000
1009	001460	000000
1010	001462	000000
1011	001464	000000
1012	001466	000000
1013	001470	000000
1014	001472	000000
1015	001474	000000
1016	001476	000000
1017		
1018		

```

$SET7: .WORD 0
$SET10: .WORD 0
$SET11: .WORD 0
$SET12: .WORD 0
NTERAD: .WORD 0
NTER0: .WORD 0
NTER1: .WORD 0
NTER2: .WORD 0
NTER3: .WORD 0
NTER4: .WORD 0
NTER5: .WORD 0
NTER6: .WORD 0
NTER7: .WORD 0
NTER10: .WORD 0
NTER11: .WORD 0
NTER12: .WORD 0
NEWSTK: .WORD 0
:END OF USER DEFINED TAGS:

```

```

: THESE LOCATIONS CONTAIN THE
: APPROPRIATE INTERLEAVE FACTORS
: FOR THE PARITY CONTROL
: REGISTERS (IF ANY)

```

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

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

1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074

001500

001500 013752
001502 015005

001504 015142
001506 000000

001510 013776
001512 014470

001514 015106
001516 000000

001520 014027
001522 014604

001524 015124
001526 000000

001530 014055
001532 014540
001534 015114
001536 000000

001540 014120
001542 014470

001544 015106
001546 000000

001550 014145
001552 014566
001554 015120
001556 000000

001560 014212

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

: ITEM 1
 EM1 : TEST DIDN'T ABORT
 DH5 : PROGRAM PC
 : REGISTER UNDER TEST
 : EXPECTED ABORT PC
 DT5 : \$HLTAD, PARITY, \$GDDAT
 0

: ITEM 2
 EM2 : FATAL ERROR TO PROGRAM
 DH1 : PROGRAM PC
 : REGISTER UNDER TEST
 DT1 : \$HLTAD, PARITY
 0

: ITEM 3
 EM3 : ABORTED INCORRECTLY
 DH4 : PROGRAM PC
 : REGISTER UNDER TEST
 : EXPECTED BITS 5 THRU 11
 : ACTUAL BITS 5 THRU 11
 : EXPECTED ABORT PC
 : ACTUAL ABORT PC
 DT4 : \$HLTAD, PARITY, \$GDADR, \$BDADR, \$GDDAT, \$BDDAT
 0

: ITEM 4
 EM4 : NO PARITY MEMORY FOUND BELOW 29K
 DH2 : REGISTER UNDER TEST
 DT2 : PARITY
 0

: ITEM 5
 EM5 : RESET DOESN'T WORK
 DH1 : PROGRAM PC
 : REGISTER UNDER TEST
 DT1 : \$HLTAD, PARITY
 0

: ITEM 6
 EM6 : USER SELECTED REGISTER NOT PRESENT
 DH3 : PROGRAM PC
 DT3 : \$HLTAD
 0

: ITEM 7
 EM7 : NO PARITY MEMORY FOUND AT ALL

1075	001562	014540
1076	001564	015114
1077	001566	000000
1078		
1079	001570	014252
1080		
1081	001572	015005
1082		
1083		
1084	001574	015142
1085	001576	000000
1086		
1087	001600	014326
1088		
1089	001602	014470
1090		
1091	001604	015106
1092	001606	000000
1093		
1094	001610	014403
1095		
1096	001612	014470
1097		
1098	001614	015106
1099	001616	000000

	DH2
	DT2
	0
; ITEM 10	
	EM10
	DH5
	DT5
	0
; ITEM 11	
	EM11
	DH1
	DT1
	0
; ITEM 12	
	EM12
	DH1
	DT1
	0

```

;REGISTER UNDER TEST
;PARITY

;DIDN'T ABORT OR RECOGNIZE
;STACK VIOLATION
;PROGRAM PC
;REGISTER UNDER TEST
;EXPECTED ABORT PC
;SHLTAD, PARITY, $GDDAT

;ABORTED BUT STACK VIOLATION
;NOT RECOGNIZED
;PROGRAM PC
;REGISTER UNDER TEST
;SHLTAD, PARITY

;STACK VIOLATION PICKED UP BLT
;ABORT NOT RECOGNIZED
;PROGRAM PC
;REGISTER UNDER TEST
;SHLTAD, PARITY

```

1100
1101
1102
1103
1104
1105
1106
1107
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151

SYSTEM PARITY REGISTER NOTES FOR MF11 AND MS11

BIT ASSIGNMENTS FOR THE MF11 PARITY REGISTER IS AS FOLLOWS:

BIT15	PARITY ERROR	
BITS 11-5	ERROR ADDRESS	; HIGH ORDER ADDRESS BITS
		; OF ADDRESS OF MOST RECENT ERROR
		; (BITS 17 THRU 11)
BIT02	WRITE	; NORMAL PARITY (ODD) WHEN CLEAR
		; OTHER PARITY (EVEN) WHEN SET
BIT00	ERROR ACTION ENABLE	; NO ACTION WHEN CLEAR
		; TRAP TO VECTOR 114 WHEN SET

NOTE: THE ABOVE BITS ARE READ/WRITE AND CAN BE CLEARED BY 'INIT' (EXCEPT BITS 11-5)

////////////////////////////////////

BIT ASSIGNMENTS FOR THE MS11 PARITY REGISTER IS AS FOLLOWS:

BIT15	PARITY ERROR	
BIT02	WRITE	; NORMAL PARITY (EVEN) WHEN CLEAR
		; OTHER PARITY (ODD) WHEN SET
BIT00	ERROR ACTION ENABLE	; NO ACTION WHEN CLEAR
		; TRAP TO VECTOR 114 WHEN SET

NOTE: THERE ARE NO ERROR ADDRESS BITS IN THE CURRENT MS11 PARITY REGISTER
HOWEVER, THERE WILL BE IN A LATER VERSION WHICH WILL BE
HANDLED PROPERLY BY THIS PROGRAM

SPECIAL NOTE----THERE ARE 2 GENERAL PURPOSE REGISTERS USED IN THE
PROGRAM FOR SPECIFIC CIRCUMSTANCES. THEY ARE:

R1 - WILL ALWAYS CONTAIN THE 1ST ADDRESS OF THE 2
LOCATION MAP USED FOR TESTING.
THE CONTENTS OF R1 IS DETERMINED BY THE 'COMPUT'
ROUTINE SHOWN FURTHER DOWN.
EXAMINATION OF R1 WILL TELL YOU WHERE IN PARITY
MEMORY TESTING IS BEING CONDUCTED.

R5 - WILL ALWAYS CONTAIN THE ADDRESS OF THE ROUTINE
FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS.

1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205

001620 000114
001622 000000
001624 000000

001626 000000
001630 000000

001632 000000

001634 000000

001636 000000

001640 000000
001642 000000

001644 000003 000001
001644 000003 000001
001650 000006 000002
001654 000011 000003
001660 000014 000004
001664 000017 000005
001670 000022 000006
001674 000025 000007
001700 000030 000010
001704 000000

```
*****
: MISCELLANEOUS COMMON PARITY VARIABLES AND FLAGS
*****
INTVEC: 114      ; PARITY INTERRUPT VECTOR ADDRESS
PARITY: 0       ; CONTAINS PARITY REGISTER IN USE
PSPCORZONES: 0  ; FLAG TO INDICATE TO 'CHECKLOC'
                 ; ROUTINE THAT A PS OR PC FETCH
                 ; OR A ZONE ABORT WAS DONE
                 ; 0 = NO; 1 = YES
MSREGFLAG: 0    ; INDICATES PARITY TYPE
                 ; 0 = CORE ; 1 = MOS
USERTYPE: 0     ; INDICATES USER SELECTION OF
                 ; PARITY REGISTER
                 ; 0 = PROGRAM FIND
                 ; 1 = USER SELECTION
BLKCNT: 0       ; CONTAINS THE NUMBER OF CONSEC-
                 ; UTIVE LOCATIONS TO BE TESTED
                 ; DURING PROGRAM TABULATION TO
                 ; COVER CASES OF MEMORY INTER-
                 ; LEAVING
RESTOREBASE: 0  ; HOLDS PAGE 1 ADDRESS
                 ; OR CURRENT MEMORY ADDRESS FOR
                 ; RESTORATION DURING
                 ; RUNNING OF PROGRAM. IT IS USED IF
                 ; WE HAVE CHECKED CONSECUTIVE
                 ; LOCATIONS WITHOUT AN ABORT BEFORE
                 ; GOING TO NEXT OFFSET WHICH WILL
                 ; PUT US IN ANOTHER BANK
LEAFCNT: 0      ; CONTAINS THE NO. OF ABORTS
                 ; ENCOUNTERED IN DETERMINATION OF
                 ; AN INTERLEAVE FACTOR
MEMAD: 0        ; CONTAINS A BASE ADDRESS OR A
                 ; CURRENT MEMORY ADDRESS USED IN
CPU4C: 0        ; FLAG TO INDICATE PROCESSOR
                 ; 0 = 11/45; 1 = 11/40
                 ; PARITY TABLE CREATION

: THE FOLLOWING TABLE IS USED TO DETERMINE THE
: INTERLEAVE FACTOR FOR THE CONTROL REGISTERS
: INTERTABLE:
3,1      ; 3 ABORTS ON 3 CONSECUTIVE LOCS. = 1 WAY LEAVE
6,2      ; 3 ABORTS ON 6 CONSECUTIVE LOCS. = 2 WAY LEAVE
9,3      ; 3 ABORTS ON 9 CONSECUTIVE LOCS. = 3 WAY LEAVE
12,4     ; 3 ABORTS ON 12 CONSECUTIVE LOCS. = 4 WAY LEAVE
15,5     ; 3 ABORTS ON 15 CONSECUTIVE LOCS. = 5 WAY LEAVE
18,6     ; 3 ABORTS ON 18 CONSECUTIVE LOCS. = 6 WAY LEAVE
21,7     ; 3 ABORTS ON 21 CONSECUTIVE LOCS. = 7 WAY LEAVE
24,8     ; 3 ABORTS ON 24 CONSECUTIVE LOCS. = 8 WAY LEAVE
0        ; END OF TABLE TERMINATOR
```

```

1206
1207 001706          BEGIN:
1208 001706 012706 001100      MOV      #STACK, SP          ; SETUP THE STACK POINTER
1209 001712 012737 012062 000020  MOV      #SCOPE, @IOTVEC      ; IOT VECTOR FOR SCOPE ROUTINE
1210 001720 012737 000340 000022  MOV      #340, @IOTVEC+2     ; LEVEL 7
1211 001726 005067 177350      CLR      $STNM              ; INITIALIZE THE TEST NUMBER
1212 001732 012737 012634 000030  MOV      #HLT, @EMTVEC       ; EMT VECTOR FOR HLT(ERROR) ROUTINE
1213 001740 012737 000340 000032  MOV      #340, @EMTVEC+2     ; LEVEL 7
1214 001746 012737 013556 000034  MOV      #TRAP, @TRAPVEC     ; TRAP VECTOR FOR TRAP CALLS
1215 001754 012737 000340 000036  MOV      #340, @TRAPVEC+2   ; LEVEL 7
1216 001762 012737 013606 000024  MOV      #SPWRDN, @PWRVEC    ; POWER FAILURE VECTOR
1217 001770 012737 000340 000026  MOV      #340, @PWRVEC+2    ; LEVEL 7
1218 001776 005067 177276      CLR      $PASS             ; CLEAR THE PASS COUNT
1219 002002 005067 177276      CLR      $ICNT             ; INITIALIZE THE ITERATION COUNTER
1220 002006 005067 010276      CLR      $TIMES            ; INITIALIZE NUMBER OF ITERATIONS
1221 002012 005067 177276      CLR      $ERFLG           ; CLEAR THE ERROR FLAG
1222 002016 005067 177270      CLR      $ERTTL           ; CLEAR THE ERROR COUNT
1223 002022 005067 010746      CLR      $ESCAPE          ; CLEAR THE ESCAPE ON ERROR ADDRESS
1224
1225 002026 005037 001630      CLR      @USERTYPE         ; SET USER SELECTION INDICATOR
1226                                     ; TO ZERO INDICATING PROGRAM
1227                                     ; TABULATION
1228 002032 005037 002304      CLR      @SKT11            ; CLEAR KT11 PRESENCE FLAG
1229 002036 005037 001420      CLR      @SSET0            ; CLEAR THE OFFSET
1230 002042 005037 001422      CLR      @SSET1            ; TABLE LOCATIONS FOR
1231 002046 005037 001424      CLR      @SSET2            ; THE KT11 OPTION
1232 002052 005037 001426      CLR      @SSET3
1233 002056 005037 001430      CLR      @SSET4
1234 002062 005037 001432      CLR      @SSET5
1235 002066 005037 001434      CLR      @SSET6
1236 002072 005037 001436      CLR      @SSET7
1237 002076 005037 001440      CLR      @SSET10
1238 002102 005037 001442      CLR      @SSET11
1239 002106 005037 001444      CLR      @SSET12
1240 002112 005037 001450      CLR      @NTER0            ; CLEAR THE INTERLEAVE TABLE
1241 002116 005037 001452      CLR      @NTER1            ; ENTRY LOCATIONS
1242 002122 005037 001454      CLR      @NTER2
1243 002126 005037 001456      CLR      @NTER3
1244 002132 005037 001460      CLR      @NTER4
1245 002136 005037 001462      CLR      @NTER5
1246 002142 005037 001464      CLR      @NTER6
1247 002146 005037 001466      CLR      @NTER7
1248 002152 005037 001470      CLR      @NTER10
1249 002156 005037 001472      CLR      @NTER11
1250 002162 005037 001474      CLR      @NTER12
1251 002166 005037 001636      CLR      @LEAFCNT          ; CLEAR NO. OF ABORTS PER NO. OF
1252                                     ; CONSECUTIVE LOCS. TESTED FLAG
1253 002172 005037 001642      CLR      @CPU40            ; CLEAR PROCESSOR INDICATOR FLAG
1254 002176 013746 000004      MOV      @4, -(SP)         ; SAVE CONTENTS OF LOC. 4
1255 002202 013746 000010      MOV      @10, -(SP)        ; SAVE CONTENTS OF LOC. 10
1256 002206 012737 002226 000010  MOV      #15, @RESVEC       ; SET UP FOR 'SPL' TRAP ADDRESS
1257 002214 012737 000340 000012  MOV      #340, @RESVEC+2    ; SET UP FOR 'SPL' TRAP PS
1258 002222 000237          SPL      7                  ; ATTEMPT TO SET A PRIORITY LEVEL
1259 002224 000403          BR       2$                ; BRANCH INDICATING WE ARE ON AN
1260                                     ; 11/45 PROCESSOR
1261 002226 022626          1$:  CMP      (SP)+, (SP)+    ; RESET THE STACK FROM TRAP

```

```

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

```

```

1318 002372 104400 002400      TYPE      +4      ;TYPE ASCIZ STRING
1319 002376 000433      BR          64$      ;GET OVER THE ASCII
1320      ;.ASCIZ      <15><12>"TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN "
1321 002466      64$:
1322 002456 104406 0C1340      ACCEPT,$REGD      ;PICK UP THE DESIRED REGISTER
1323      ;FROM THE TELETYPE AND STORE
1324      ;IN FIRST TABLE LOCATION
1325 002472 005237 001637      INC      @#USERTYPE      ;SET FLAG INDICATING USER SELECTION
1326 002476 013746 000004      MOV      @#4,-(SP)      ;SAVE CONTENTS OF LOC. 4
1327 002502 000404      BR          NEXT1      ;SKIP THE NEXT INSTRUCTIONS
1328 002504 012712 172100      FINDONE:  MOV      #172100,(R2) ;MOVE 1ST POSSIBLE PARITY
1329      ;REGISTER INTO $REGD
1330 002510 013746 000004      MOV      @#4,-(SP)      ;PUSH CONTENTS OF LOC.4 ONTO STACK
1331 002514 012704 001644      NEXT1:  MOV      #INTERTABLE,R4 ;INITIALIZE INTERLEAVE TABLE
1332      ;POINTER
1333 002520 005737 002304      TST      @#$KT11      ;KT11 ARE YOU THERE?
1334 002524 001402      BEQ      5$           ;BRANCH IF NO
1335 002526 005077 175500      CLR      @KPAR1      ;RESET PAGE 1 ADDRESS REGISTER
1336      ;BEFORE TESTING NEXT PARITY
1337      ;CONTROL REGISTER
1338 002532 012737 003144 000004 5$:  MOV      #NOREG,@#4      ;SET PARITY TIMEOUT VECTOR SERVICE ADDRESS
1339 002540 022712 172136      CMP      #172136,(R2)   ;IS THE ADDRESS IN BOUNDS?
1340 002544 100002      BPL      12$         ;BRANCH IF YES
1341 002546 000137 003172      JMP      @#NOMORE      ;OTHERWISE - TERMINATE TABLE!
1342 002552 005772 000000      12$:  TST      @ (R2)      ;YES - IS THIS REGISTER PRESENT?
1343 002556 004737 003222      JSR      R7,@#PARTST   ;CHECK OUT FOR FATAL ERRORS
1344
1345
1346      ;WE HAVE CHECKED OUT THE REGISTER AND FOUND IT TO BE WORKING PROPERLY
1347      ;NOW WE WILL FIND ITS ASSOCIATED PARITY MEMORY, IF IT EXISTS!!
1348
1349
1350 002562 012737 003112 000004      MOV      #PARCORE,@#4   ;SET MEMORY TIMEOUT VECTOR
1351      ;SERVICE ADDRESS
1352 002570 012737 013700 001640      MOV      #13700,@#MEMAD ;SET UP A STARTING ADDRESS
1353 002576 011437 001632      MOV      (R4),@#BLKCNT  ;SET A COUNTER FOR CONSECUTIVE
1354      ;LOCATION CHECKS TO COVER MEMORY
1355      ;INTERLEAVING
1356 002602 005737 002304      TST      @#$KT11      ;SHOULD I LOOK ABOVE 28K?
1357 002606 100013      BPL      1$           ;BRANCH IF NO
1358 002610 062737 010000 001640      ADD      #10000,@#MEMAD ;STEP UP TO A PAGE 1 BASE ADDRESS
1359      ;IF MEMORY MANAGEMENT TURNED ON
1360 002616 013737 001640 001634      MOV      @#MEMAD,@#RESTOREBASE ;SAVE PAGE 1 BASE ADDRESS
1361 002624 062710 000140      2$:  ADD      #140,(R0)      ;SET UP AN OFFSET FOR KPAR1
1362 002630 011077 175376      MOV      (R0),@KPAR1   ;SET OFFSET IN PAGE 1 REGISTER
1363 002634 000406      BR          9$         ;SKIP NEXT 2 INSTRUCTIONS
1364 002636 062737 004000 001640 1$:  ADD      #4000,@#MEMAD  ;STEP UP TO NEXT BANK
1365 002644 013737 001640 001634      MOV      @#MEMAD,@#RESTOREBASE ;SAVE INITIAL MEMORY ADDRESS
1366 002652 005777 176762      9$:  TST      @MEMAD      ;IS THIS MEMORY AVAILABLE?
1367 002656 013713 001640      MOV      @#MEMAD,(R3)  ;YES - STORE THIS MEMORY LOCATION
1368 002662 004737 003430      JSR      R7,@#ABORT    ;NOW LET'S SEE IF IT'S PARITY
1369      ;MEMORY CORRESPONDING TO THE
1370      ;PARITY REGISTER WE'VE FOUND
1371 002666 005737 002304      TST      @#$KT11      ;KT11 ARE YOU THERE?
1372 002672 100034      BPL      4$           ;BRANCH IF NO
1373 002674 005337 001632      DEC      @#BLKCNT      ;DECREASE CONSECUTIVE

```



```

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

```



```

1542 ;*****
1543 ;TEST 5 WHICH OPTION IS ABOUT TO BE TESTED
1544 ;*****
1545 003374 000004 TST5: SCOPE
1546
1547 003376 052777 007740 176216 WHICH1: BIS #7740, @PARITY ;IS AN OLD MS11 OPTION
1548 ;WITH NO ADDRESS BITS
1549 ;ABOUT TO BE TESTED?
1550 003404 032777 007740 176210 BIT #7740, @PARITY ;ADDRESS BITS ABLE TO BE SET?
1551 003412 001402 BEQ 15 ;BRANCH IF NO INDICATING MS11
1552 003414 000004 SCOPE
1553 003416 000207 RTS R7 ;RETURN TO NORMAL FLOW
1554 003420 005237 001626 15: INC @#MSREGFLAG ;SET FLAG INDICATING MS11 OPTION
1555 ;WITH NO ADDRESS BITS
1556 003424 000004 SCOPE
1557 003426 000207 RTS R7 ;RETURN TO NORMAL FLOW
1558
1559 ;*****
1560 ;
1561 ;THE FOLLOWING ROUTINE WILL TAKE EACH 1K BANK OF MEMORY
1562 ;THAT IS AVAILABLE AND PERFORM A DATI IN IT
1563 ;TO DETERMINE IF PARITY EXISTS THERE. THIS ROUTINE IS
1564 ;ONLY USED DURING TABLE CREATION
1565 ;
1566 ;*****
1567 003430 010546 ABORT: MOV R5, -(SP) ;SAVE R5 CONTENTS ON STACK
1568 003432 010046 MOV R0, -(SP) ;SAVE R0 CONTENTS ON STACK
1569 003434 011300 MOV (R3), R0 ;GET THE MEMORY LOCATION
1570 ;JUST DETERMINED
1571 003436 004337 003204 JSR R3, @#COMPUT ;COMPUTE AN AREA IN THIS BANK
1572 ;FOR DETERMINING PARITY MEMORY
1573 003442 011267 176154 MOV (R2), PARITY ;GET THE PARITY REGISTER JUST
1574 ;FOUND AND TEST WITH IT
1575 ;*****
1576 ;TEST A DATI IN THIS BANK
1577 ;*****
1578 ;11/45 **** ROM STATE 221 ****
1579 ;
1580 ;11/40 **** ROM STATE 207 ****
1581 003446 012705 011450 MOV #VECSET, R5 ;SET UP SERVICE ROUTINE ADDRESS
1582 003452 004015 JSR R0, (R5) ;SET UP PARITY VECTOR SERVICE
1583 003454 003500 ONETRY ;ROUTINE ADDRESS
1584 003456 011100 MOV @R1, R0 ;SET UP FOR A DATO
1585 003460 010010 MOV R0, @R0 ;DO THE DATO
1586 003462 010030 MOV R0, @R0+ ;DO A DATI
1587 003464 042777 000005 176130 BIC #BIT2!BIT0, @PARITY ;WRITE NORMAL AND DISABLE
1588 003472 012600 MOV (SP)+, R0 ;RESTORE R0 CONTENTS
1589 003474 012605 MOV (SP)+, R5 ;RESTORE R5 CONTENTS
1590 003476 000207 RTS R7 ;NOT PARITY MEMORY!
1591 ;RETURN TO TEST AT NEXT
1592 ;INCREMENT
1593 003500 042777 000005 176114 ONETRY: BIC #BIT2!BIT0, @PARITY ;WE HAVE PARITY MEMORY - PROCEED
1594 003506 016600 000004 MOV 4(SP), R0 ;RESTORE R0 CONTENTS
1595 003512 016605 000006 MOV 6(SP), R5 ;RESTORE R5 CONTENTS
1596 003516 005237 001636 INC @#LEAFCNT ;INCREMENT INTERLEAVE COUNTER
1597 003522 022737 000003 001636 CMP #3, @#LEAFCNT ;3 ABORTS REACHED?

```

Z

```

1598 003530 001403          BEQ      1$          ;BRANCH IF YES
1599 003532 062706 000010  ADD      #10,SP     ;BYPASS JUNK ON STACK
1600 003535 000207          RTS      R7         ;RETURN TO TEST AT NEXT INCREMENT
1601 003540 022626          1$:  CMP      (SP)+,(SP)+ ;POP STACK BACK FROM PARITY ABORT
1602 003542 012600          MOV      (SP)+,R0   ;RESTORE R0 CONTENTS
1603 003544 012605          MOV      (SP)+,R5   ;RESTORE R5 CONTENTS
1604 003546 005726          TST      (SP)+     ;POP STACK ONCE FOR ABORT ROUTINE
1605                                     ;ENTRY
1606 003550 005737 001630  TST      @#USERTYPE ;DID USER TYPE IN REGISTER?
1607 003554 001404          BEQ      2$          ;BRANCH IF NO
1608 003556 016415 000002  MOV      2(R4),(R5) ;SET INTERLEAVE VALUE INTO
1609                                     ;TABLE
1610 003552 000137 003620  JMP      @#START    ;AND LOCK ON THE USER SELECTED
1611                                     ;REGISTER FOR TESTING
1612 003565 005723          2$:  TST      (R3)+   ;USER DIDN'T SELECT - SO STEP UP
1613                                     ;TO NEXT MEMORY TABLE LOCATION
1614 003570 005720          TST      (R0)+     ;STEP UP TO NEXT OFFSET TABLE
1615                                     ;LOCATION - THIS TABLE WILL ONLY
1616                                     ;BE APPLICABLE IF MEMORY MGMT
1617                                     ;IS TURNED ON
1618 003572 012212          MOV      (R2)+,(R2) ;SET NEXT POSSIBLE REGISTER INTO
1619 003574 062712 000002  ADD      #2,(R2)   ;NEXT REGISTER TABLE LOCATION
1620 003600 016425 000002  MOV      2(R4),(R5)+ ;SET INTERLEAVE VALUE INTO
1621                                     ;TABLE
1622 003604 005037 001636  CLR      @#LEAFCNT ;RESET NO. OF ABORTS COUNTER
1623 003610 005037 001626  CLR      @#MSREGFLAG ;CLEAR PARITY TYPE INDICATOR
1624 003614 000137 002514  JMP      @#NEXT1   ;GO BACK TO CHECK NEXT POSSIBLE
1625                                     ;PARITY REGISTER

```

```

*****
*****
*****
*****
*****

```

```

: IF WE HAVE REACHED THIS POINT IN THE PROGRAM THEN,
: $REGD(LOCATION 1340) WILL CONTAIN THE FIRST PARITY REGISTER
:
: LOCATION 1342 UP TO 1364 WILL CONTAIN ANY MORE
: REGISTERS FOUND
: $TMPD(LOCATION 1366) WILL CONTAIN AN ADDRESS IN THE BANK
: THAT HAS PARITY ASSOCIATED WITH THE PARITY
: REGISTER IN $REGD
:
: LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING
: MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS
: $SETD(LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED
: WITH THE CORRESPONDING VALUE IN $TMPD
:
: LOCATION 1422 TO 1444 WILL CONTAIN THE CORRESPONDING
: OFFSET VALUES FOR THE OTHER REGISTERS
:
: $INTERD(LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED

```

1653

1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709

003620 005777 175512
003624 001401
003626 000402
003630 000137 011464
003634 017767 175476 175760
003642 162767 000002 175432

003650 005737 002304
003654 100906
003656 017777 175534 174346
003664 062767 000001 175410

003672 017700 175516
003676 004337 003204
003702 010137 001476

003706 004737 003376
003712 012705 011450

003716 000004

003720 004015

WITH THE PARITY REGISTER IN \$REGO
LOCATION 1452 TO 1474 WILL CONTAIN THE CORRESPONDING
INTERLEAVE FACTORS FOR THE OTHER REGISTERS
NOTE: TO UNDERSTAND THE TABLE SETUP SEE THE DOCUMENT

THE REST OF THE PROGRAM WILL TEST ALL PARITY ABORTS
USING EITHER THE PARITY REGISTER TYPED BY THE USER OR THOSE
ENCOUNTERED IN THE TABLE PREVIOUSLY GENERATED.

START: TST \$REGAD ; ARE WE AT END OF TABLE?
BEQ 15 ; BRANCH IF YES
BR 25 ; OTHERWISE SKIP NEXT INSTRUCTION
15: JMP \$RESTART ; SET UP TO START TABLE OVER!!
25: MOV \$REGAD, PARITY ; GET PARITY REGISTER TO BE USED
SUB #2, \$STNM ; DECREASE TEST NO. FOR SCOPE
; LOOPING IN CASE TEST #4 WAS
; DONE BECAUSE TEST #4 BEING
; DONE THROWS \$STNM COUNT OFF
; BY 2
TST \$KT11 ; KT11 ARE YOU THERE?
BPL 35 ; BRANCH IF NO
MOV \$SETAD, \$KPAR1 ; GET THE OFFSET NEEDED
ADD #1, \$STNM ; SET TEST NO. TO A CORRECT
; VALUE; TEST #4 WAS NOT
; EXECUTED
35: MOV \$TMPAD, R0 ; GET PARITY MEMORY ASSOCIATED
; WITH THE PARITY REGISTER
JSR R3, \$COMPUT ; COMPUTE AN AREA IN THIS BANK
; FOR TESTING
MOV R1, \$NEWSTK ; SET UP A NEW STACK POINTER
; FOR STACK OPERATIONS IN CASE
; NO PARITY MEMORY RESIDES IN
; LOWER 4K
JSR PC, \$WHICH1 ; DETERMINE IF WE ARE ABOUT TO
; TEST AN OLD MOS DESIGN!
MOV \$VECSET, R5 ; SET UP SERVICE ROUTINE ADDRESS

; TEST 6 TEST (ADDRESS) SMO, DM3 MOV INSTRUCTION

TST6: SCOPE
; 11/45 **** ROM STATE 221 ****
; 11/40 **** ROM STATE 207 ****
JSR R0, (R5) ; SET UP PARITY VECTOR SERVICE

```

1710 003722 003752      A      ;ROUTINE ADDRESS
1711 003724 011100      MOV    @R1,RO      ;SET UP FOR DATO
1712 003726 010010      MOV    RO,@RO      ;DO THE DATO
1713 003730 012737 003740 001332  MOV    @.+10,@#$GDDAT ;STORE THE PC THAT SHOULD
1714                                     ;BE PUSHED ON THE STACK
1715                                     ;IF A PARITY ABORT OCCURS
Z 1716 003736 010030      MOV    RO,@(RO)+    ;DO A DATI
1717 003740 042777 000004 175654  BIC    #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1718 003746 104001      HLT    +1          ;DIDN'T ABORT
1719 003750 000410      BR     .+22        ;GO TO NEXT TEST
1720 003752 042777 000005 175642 A:    BIC    #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1721 003760 004037 011550      JSR    RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
1722 003754 104003      HLT    +3          ;ABORTED INCORRECTLY
1723 003766 012706 001100      MOV    #STACK,SP   ;RESET THE STACK
1724                                     ;*****
1725 ;TEST 7 TEST (ADDRESS) SMO,DM5 MOV INSTRUCTION
1726                                     ;*****
1727 003772 000004      ST7:  SCOPE
1728                                     ;
1729                                     ;
1730                                     ;
1731                                     ;
1732                                     ;
1733                                     ;
1734                                     ;
1735 004004 012737 004014 001772  MOV    @.+10,@#$GDDAT ;STORE THE PC THAT SHOULD
1736                                     ;BE PUSHED ON THE STACK
1737                                     ;IF A PARITY ABORT OCCURS
Z 1738 004012 010050      MOV    RO,@-(RO)   ;DO A DATI
1739 004014 042777 000004 175600  BIC    #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1740 004022 104001      HLT    +1          ;DIDN'T ABORT
1741 004024 000410      BR     .+22        ;GO TO NEXT TEST
1742 004026 042777 000005 175566 A0:   BIC    #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1743 004034 004037 011550      JSR    RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
1744 004040 104003      HLT    +3          ;ABORTED INCORRECTLY
1745 004042 012706 001100      MOV    #STACK,SP   ;RESET THE STACK
1746                                     ;*****
1747 ;TEST 10 TEST (DATA) SM1,DM6 MOV INSTRUCTION
1748                                     ;*****
1749 004046 000004      ST10: SCOPE
1750                                     ;
1751                                     ;
1752                                     ;
1753                                     ;
1754                                     ;
1755                                     ;
1756                                     ;
1757 004060 012737 004070 001332  MOV    @.+10,@#$GDDAT ;STORE THE PC THAT SHOULD
1758                                     ;BE PUSHED ON THE STACK
1759                                     ;IF A PARITY ABORT OCCURS
1760 004066 011060 177776      MOV    @RO,-2(RO)  ;DO A DATI
1761 004072 042777 000004 175522  BIC    #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1762 004100 104001      HLT    +1          ;DIDN'T ABORT
1763 004102 000410      BR     .+22        ;GO TO NEXT TEST
1764 004104 042777 000005 175510 A1:   BIC    #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1765 004112 004037 011550      JSR    RO,@#CHECKLOC ;CHECK FOR GOOD ABORT

```

```

1766 004116 104003          HLT      +3          ;ABORTED INCORRECTLY
1767 004120 012706 001100  MOV      #STACK,SP  ;RESET THE STACK
1768                                     ;*****
1769                                     ;TEST 11      TEST (ADDRESS) SMO,DM7  MOV INSTRUCTION
1770                                     ;*****
1771 004124 000004  TST11: SCOPE
1772                                     11/45 **** ROM STATE 231 ****
1773                                     :
1774                                     :
1775 004126 004015          JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
1776 004130 004162          A2      ;ROUTINE ADDRESS
1777 004132 011100          MOV      @R1,RO      ;SET UP FOR A DATO
1778 004134 010029          MOV      RO,(RO)+    ;DO THE DATO
1779 004136 012737 004150 001332  MOV      #.+12,@#SGDDAT ;STORE THE PC THAT SHOULD
1780                                     ;BE PUSHED ON THE STACK
1781                                     ;IF A PARITY ABORT OCCURS
1782 004144 010070 177776          MOV      RO,@-2(RO)  ;DO A DATI
1783 004150 042777 000004 175444          BIC      #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1784 004156 104001          HLT      +1          ;DIDN'T ABORT
1785 004160 000410          BR       .+22        ;GO TO NEXT TEST
1786 004162 042777 000005 175432  A2:      BIC      #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1787 004170 004037 011550          JSR      RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
1788 004174 104003          HLT      +3          ;ABORTED INCORRECTLY
1789 004176 012706 001100  MOV      #STACK,SP  ;RESET THE STACK
1790                                     ;*****
1791                                     ;TEST 12      TEST (DATA) SMO,DM2  CMP INSTRUCTION
1792                                     ;*****
1793 004202 000004  TST12: SCOPE
1794                                     11/45 **** ROM STATE 175 ****
1795                                     :
1796                                     :
1797 004204 004015          JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
1798 004206 004236          B        ;ROUTINE ADDRESS
1799 004210 011100          MOV      @R1,RO      ;SET UP FOR DATO
1800 004212 010010          MOV      RO,@RO      ;DO THE DATO
1801 004214 012737 004224 001332  MOV      #.+10,@#SGDDAT ;STORE THE PC THAT SHOULD
1802                                     ;BE PUSHED ON THE STACK
1803                                     ;IF A PARITY ABORT OCCURS
1804 004222 020020          CMP      RO,(RO)+    ;DO A DATIP, DATI
1805 004224 042777 000004 175370          BIC      #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
1806 004232 104001          HLT      +1          ;DIDN'T ABORT
1807 004234 000410          BR       .+22        ;GO TO NEXT TEST
1808 004236 042777 000005 175356  B:      BIC      #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
1809 004244 004037 011550          JSR      RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
1810 004250 104003          HLT      +3          ;ABORTED INCORRECTLY
1811 004252 012706 001100  MOV      #STACK,SP  ;RESET THE STACK
1812                                     ;*****
1813                                     ;TEST 13      TEST (DATA) SMO,DM4  CMP INSTRUCTION
1814                                     ;*****
1815 004256 000004  TST13: SCOPE
1816                                     11/45 **** ROM STATE 177 ****
1817                                     :
1818                                     :
1819 004260 004015          JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
1820 004262 004312          BO      ;ROUTINE ADDRESS
1821 004264 011100          MOV      @R1,RO      ;SET UP FOR DATO

```



```

Z 1822 004266 010020          MOV      RO,(RO)+      ;DO THE DATO
1823 004270 012737 004300 001332  MOV      #.+10,2#SGDDAT ;STORE THE PC THAT SHOULD
1824                                     ;BE PUSHED ON THE STACK
1825                                     ;IF A PARITY ABORT OCCURS
Z 1826 004276 020040          CMP      RO,-(RO)      ;DO A DATIP, DATI
1827 004300 042777 000004 175314  BIC      #BIT2,2PARITY ;WRITE NORMAL FOR EMT CALL
1828 004306 104001          HLT      +1           ;DIDN'T ABORT
1829 004310 000410          BR       .+22        ;GO TO NEXT TEST
1830 004312 042777 000005 175302  B0:     BIC      #BIT2!BIT0,2PARITY ;WRITE NORMAL AND DISABLE
1831 004320 004037 011550          JSR      RO,2#CHECKLOC ;CHECK FOR GOOD ABORT
1832 004324 104003          HLT      +3           ;ABORTED INCORRECTLY
1833 004326 012706 001100          MOV      #STACK,SP   ;RESET THE STACK
1834                                     ;*****
1835                                     ;TEST 14          TEST (DATA) SMO,DM6 CMP INSTRUCTION
1836                                     ;*****
1837 004332 000004          TST14:  SCOPE
1838                                     ;
1839                                     ;          11/45 **** ROM STATE 177 ****
1840                                     ;
1841                                     ;          11/40 **** ROM STATE 267 ****
1842 004334 004015          JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
1843 004336 004370          B1      ;ROUTINE ADDRESS
1844 004340 011100          MOV      2R1,RO      ;SET UP FOR DATO
1845 004342 010020          MOV      RO,(RO)+    ;DO THE DATO
1846 004344 012737 004356 001332  MOV      #.+12,2#SGDDAT ;STORE THE PC THAT SHOULD
1847                                     ;BE PUSHED ON THE STACK
1848                                     ;IF A PARITY ABORT OCCURS
1849 004352 020060 177776          CMP      RO,-2(RO)   ;DO A DATI, DATIP
1850 004356 042777 000004 175236  BIC      #BIT2,2PARITY ;WRITE NORMAL FOR EMT CALL
1851 004364 104001          HLT      +1           ;DIDN'T ABORT
1852 004366 000410          BR       .+22        ;GO TO NEXT TEST
1853 004370 042777 000005 175224  B1:     BIC      #BIT2!BIT0,2PARITY ;WRITE NORMAL AND DISABLE
1854 004376 004037 011550          JSR      RO,2#CHECKLOC ;CHECK FOR GOOD ABORT
1855 004402 104003          HLT      +3           ;ABORTED INCORRECTLY
1856 004404 012706 001100          MOV      #STACK,SP   ;RESET THE STACK
1857                                     ;*****
1858                                     ;TEST 15          TEST (DATA) SMO,DM1 CMP INSTRUCTION
1859                                     ;*****
1860 004410 000004          TST15:  SCOPE
1861                                     ;
1862                                     ;          11/45 **** ROM STATE 175 ****
1863                                     ;
1864                                     ;          11/40 **** ROM STATE 267 ****
1865 004412 004015          JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
1866 004414 004444          B2      ;ROUTINE ADDRESS
1867 004416 011100          MOV      2R1,RO      ;SET UP FOR DATO
1868 004420 010010          MOV      RO,2RO      ;DO THE DATO
1869 004422 012737 004432 001332  MOV      #.+10,2#SGDDAT ;STORE THE PC THAT SHOULD
1870                                     ;BE PUSHED ON THE STACK
1871                                     ;IF A PARITY ABORT OCCURS
1872 004430 020010          CMP      RO,2RO      ;DO A DATIP, DATI
1873 004432 042777 000004 175162  BIC      #BIT2,2PARITY ;WRITE NORMAL FOR EMT CALL
1874 004440 104001          HLT      +1           ;DIDN'T ABORT
1875 004442 000410          BR       .+22        ;GO TO NEXT TEST
1876 004444 042777 000005 175150  B2:     BIC      #BIT2!BIT0,2PARITY ;WRITE NORMAL AND DISABLE
1877 004452 004037 011550          JSR      RO,2#CHECKLOC ;CHECK FOR GOOD ABORT
1878 004456 104003          HLT      +3           ;ABORTED INCORRECTLY
1879 004460 012706 001100          MOV      #STACK,SP   ;RESET THE STACK

```



```

1878
1879
1880
1881 004464 000004
1882
1883
1884
1885 004466 004015
1886 004470 004536
1887 004472 011100
Z 1888 004474 010020
1889 004476 042777 000004 175116
1890 004504 011110
1891 004506 052777 000004 175106
1892 004514 012737 004524 001332
1893
1894
Z 1895 004522 020030
1896 004524 042777 000004 175070
1897 004532 104001
1898 004534 000410
1899 004536 042777 000005 175056 B3:
1900 004544 004037 011550
1901 004550 104003
1902 004552 012706 001100
1903
1904
1905
1906 004556 000004
1907
1908
1909
1910 004560 004015
1911 004562 004612
1912 004564 011100
1913 004566 010010
1914 004570 012737 004600 001332
1915
1916
Z 1917 004576 020030
1918 004600 042777 000004 175014
1919 004606 104001
1920 004610 000410
1921 004612 042777 000005 175002 B4:
1922 004620 004037 011550
1923 004624 104003
1924 004626 012706 001100
1925
1926
1927
1928 004632 000004
1929
1930
1931
1932 004634 004015
1933 004636 004706

```

```

:*****
:TEST 16      TEST (DATA) SMO,DM3  CMP INSTRUCTION
:*****
↑ST16:  SCOPE
          11/45 **** ROM STATE 177 ****
:
:          11/40 **** ROM STATE 267 ****
:          JSR      RO,(R5)          ;SET UP PARITY VECTOR SERVICE
:          B3          ;ROUTINE ADDRESS
:          MOV      @R1,RO          ;SET UP FOR DATO
:          MOV      RO,@RO+        ;DO THE DATO
:          BIC      #BIT2,@PARITY  ;WRITE NORMAL
:          MOV      @R1,@RO        ;WRITE ADDRESS NORMAL (DATI)
:          BIS      #BIT2,@PARITY  ;WRITE OTHER PARITY
:          MOV      #.+10,@#SGDDAT ;STORE THE PC THAT SHOULD
:                               ;BE PUSHED ON THE STACK
:                               ;IF A PARITY ABORT OCCURS
:          CMP      RO,@(RO)+      ;DO A DATI, DATIP
:          BIC      #BIT2,@PARITY  ;WRITE NORMAL FOR EMT CALL
:          HLT      +1             ;DIDN'T ABORT
:          BR       .+22          ;GO TO NEXT TEST
:          BIC      #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
:          JSR      RO,@#CHECKLOC  ;CHECK FOR GOOD ABORT
:          HLT      +3             ;ABORTED INCORRECTLY
:          MOV      #STACK,SP      ;RESET THE STACK
:*****
:TEST 17      TEST (ADDRESS) SMO,DM3  CMP INSTRUCTION
:*****
↑ST17:  SCOPE
          11/45 **** ROM STATE 221 ****
:
:          11/40 **** ROM STATE 264 ****
:          JSR      RO,(R5)          ;SET UP PARITY VECTOR SERVICE
:          B4          ;ROUTINE ADDRESS
:          MOV      @R1,RO          ;SET UP FOR A DATO
:          MOV      RO,@RO        ;DO THE DATO
:          MOV      #.+10,@#SGDDAT ;STORE THE PC THAT SHOULD
:                               ;BE PUSHED ON THE STACK
:                               ;IF A PARITY ABORT OCCURS
:          CMP      RO,@(RO)+      ;DO A DATI
:          BIC      #BIT2,@PARITY  ;WRITE NORMAL FOR EMT CALL
:          HLT      +1             ;DIDN'T ABORT
:          BR       .+22          ;GO TO NEXT TEST
:          BIC      #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
:          JSR      RO,@#CHECKLOC  ;CHECK FOR GOOD ABORT
:          HLT      +3             ;ABORTED INCORRECTLY
:          MOV      #STACK,SP      ;RESET THE STACK
:*****
:TEST 20      TEST DATI (DATA) SMO,DM5  CMP INSTRUCTION
:*****
↑ST20:  SCOPE
          11/45 **** ROM STATE 177 ****
:
:          11/40 **** ROM STATE 267 ****
:          JSR      RO,(R5)          ;SET UP PARITY VECTOR SERVICE
:          B5          ;ROUTINE ADDRESS

```


MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 44
CPU PARITY TEST MAIN FLOW

1990	005040	000410		BR	+.22		:GO TO NEXT TEST
1991	005042	042777	000005	BIC	#BIT2:BIT0,0PARITY		:WRITE NORMAL AND DISABLE
1992	005050	004037	011550	JSR	RO,0#CHECKLOC		:CHECK FOR GOOD ABORT
1993	005054	104003		HLT	+3		:ABORTED INCORRECTLY
1994	005056	012706	001100	MOV	#STACK,SP		:RESET THE STACK
1995				;*****			
1996				;TEST 23 TEST (DATA) SM4 DMO CMP INSTRUCTION			
1997				;*****			
1998	005062	000004		TST23: SCOPE			

```

;*****
;TEST 23 TEST (DATA) SM4 DMO CMP INSTRUCTION
;*****
TST23: SCOPE

```

J04

MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 45
CPU PARITY TEST MAIN FLOW

1999

:

11/45 **** ROM STATE 27 ****

K04

MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 46
CPU PARITY TEST MAIN FLOW

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

```

2006 005074 012737 005104 001332      MOV      #.+10, @#SGDDAT      ;STORE THE PC THAT SHOULD
2007                                     ;BE PUSHED ON THE STACK
2008                                     ;IF A PARITY ABORT OCCURS
2009 005102 024000      CMP      -(RO), RO          ;DO A DATI
2010 005104 042777 000004 174510      BIC      #BIT2, @PARITY     ;WRITE NORMAL FOR EMT CALL
2011 005112 104001      HLT      +1                 ;DIDN'T ABORT
2012 005114 000410      BR       .+22              ;GO TO NEXT TEST
2013 005116 042777 000005 174476 CO:    BIC      #BIT2!BIT0, @PARITY ;WRITE NORMAL AND DISABLE
2014 005124 004037 011550      JSR      RO, @#CHECKLOC     ;CHECK FOR GOOD ABORT
2015 005130 104003      HLT      +3                 ;ABORTED INCORRECTLY
2016 005132 012706 001100      MOV      #STACK, SP        ;RESET THE STACK
2017                                     ;*****
2018                                     ;TEST 24      TEST (DATA) SM3, DMO CMP INSTRUCTION
2019                                     ;*****
2020 005136 000004      †ST24:  SCOPE
2021                                     ;
2022                                     ;          11/45 **** ROM STATE 146 ****
2023                                     ;
2024                                     ;          11/40 **** ROM STATE 250 ****
2025 005140 004015      JSR      RO, (R5)          ;SET UP PARITY VECTOR SERVICE
2026 005142 005210      C1
2027 005144 011100      MOV      @R1, RO           ;ROUTINE ADDRESS
2028 005146 010020      MOV      RO, (RO)+         ;SET UP FOR DATO
2029 005150 042777 000004 174444      BIC      #BIT2, @PARITY     ;DO THE DATO (DATA OTHER PARITY)
2030 005156 011110      MOV      @R1, @RO          ;WRITE NORMAL
2031 005160 052777 000004 174434      BIS      #BIT2, @PARITY     ;DO A DATO (ADDRESS NORMAL)
2032 005166 012737 005176 001332      MOV      #.+10, @#SGDDAT   ;WRITE OTHER PARITY
2033                                     ;STORE THE PC THAT SHOULD
2034                                     ;BE PUSHED ON THE STACK
2035 005174 023000      CMP      @-(RO)+, RO       ;IF A PARITY ABORT OCCURS
2036 005176 042777 000004 174416      BIC      #BIT2, @PARITY     ;DO A DATI
2037 005204 104001      HLT      +1                 ;WRITE NORMAL FOR EMT CALL
2038 005206 000410      BR       .+22              ;DIDN'T ABORT
2039 005210 042777 000005 174404 C1:    BIC      #BIT2!BIT0, @PARITY ;GO TO NEXT TEST
2040 005216 004037 011550      JSR      RO, @#CHECKLOC     ;WRITE NORMAL AND DISABLE
2041 005222 104003      HLT      +3                 ;CHECK FOR GOOD ABORT
2042 005224 012706 001100      MOV      #STACK, SP        ;ABORTED INCORRECTLY
2043                                     ;RESET THE STACK
2044                                     ;*****
2045                                     ;TEST 25      TEST (DATA) SMS, DMO CMP INSTRUCTION
2046                                     ;*****
2047 †ST25:  SCOPE
2048                                     ;
2049                                     ;          11/45 **** ROM STATE 146 ****
2050                                     ;
2051                                     ;          11/40 **** ROM STATE 250 ****
2052 005232 004015      JSR      RO, (R5)          ;SET UP PARITY VECTOR SERVICE
2053 005234 005304      C2
2054 005236 011100      MOV      @R1, RO           ;ROUTINE ADDRESS
2055 005240 042777 000004 174354      BIC      #BIT2, @PARITY     ;SET UP FOR A DATO
2056 005246 010060 177776      MOV      RO, -2(R0)        ;WRITE NORMAL
2057 005252 052777 000004 174342      BIS      #BIT2, @PARITY     ;DO A DATO (ADDRESS NORMAL)
2058 005260 011110      MOV      @R1, @RO          ;WRITE OTHER PARITY
2059 005262 012737 005272 001332      MOV      #.+10, @#SGDDAT   ;DO A DATO (DATA OTHER PARITY)
2060                                     ;STORE THE PC THAT SHOULD
2061                                     ;BE PUSHED ON THE STACK
2062                                     ;IF A PARITY ABORT OCCURS
2063 005270 025000      CMP      @-(RO), RO       ;DO A DATI
2064 005272 042777 000004 174322      BIC      #BIT2, @PARITY     ;WRITE NORMAL FOR EMT CALL
2065 005300 104001      HLT      +1                 ;DIDN'T ABORT

```

Z

```

2062 005302 000410          BR      .+22          ;GO TO NEXT TEST
2063 005304 042777 000005 174310 C2:  BIC      #BIT2!BIT0, @PARITY ;WRITE NORMAL AND DISABLE
2064 005312 004037 011550          JSR      RO, @#CHECKLOC ;CHECK FOR GOOD ABORT
2065 005316 104003          HLT      +3          ;ABORTED INCORRECTLY
2066 005320 012706 001100          MOV      #STACK, SP ;RESET THE STACK
2067                                     ;*****
2068                                     ;TEST 26          TEST (DATA) SM1, DMO CMP INSTRUCTION
2069                                     ;*****
2070 005324 000004          †ST26: SCOPE
2071                                     ;          11/45 **** ROM STATE 27 ****
2072                                     ;
2073                                     ;          11/40 **** ROM STATE 250 ****
2074 005326 004015          JSR      RO, (R5) ;SET UP PARITY VECTOR SERVICE
2075 005330 005360          C3          ;ROUTINE ADDRESS
2076 005332 011100          MOV      @R1, RO ;SET UP FOR DATO
2077 005334 010010          MOV      RO, @RO ;DO THE DATO
2078 005336 012737 005346 001332          MOV      #.+10, @#$GDDAT ;STORE THE PC THAT SHOULD
2079                                     ;BE PUSHED ON THE STACK
2080                                     ;IF A PARITY ABORT OCCURS
2081 005344 021000          CMP      @RO, RO ;DO A DATI
2082 005346 042777 000004 174246          BIC      #BIT2, @PARITY ;WRITE NORMAL FOR EMT CALL
2083 005354 104001          HLT      +1          ;DIDN'T ABORT
2084 005356 000410          BR      .+22          ;GO TO NEXT TEST
2085 005360 042777 000005 174234 C3:  BIC      #BIT2!BIT0, @PARITY ;WRITE NORMAL AND DISABLE
2086 005366 004037 011550          JSR      RO, @#CHECKLOC ;CHECK FOR GOOD ABORT
2087 005372 104003          HLT      +3          ;ABORTED INCORRECTLY
2088 005374 012706 001100          MOV      #STACK, SP ;RESET THE STACK
2089                                     ;*****
2090                                     ;TEST 27          TEST (DATA) SM6, DMO CMP INSTRUCTION
2091                                     ;*****
2092 005400 000004          †ST27: SCOPE
2093                                     ;          11/45 **** ROM STATE 142 ****
2094                                     ;
2095                                     ;          11/40 **** ROM STATE 250 ****
2096 005402 004015          JSR      RO, (R5) ;SET UP PARITY VECTOR SERVICE
2097 005404 005436          C4          ;ROUTINE ADDRESS
2098 005406 011100          MOV      @R1, RO ;SET UP FOR A DATO
2099 005410 010020          MOV      RO, (RO)+ ;DO THE DATO
2100 005412 012737 005424 001332          MOV      #.+12, @#$GDDAT ;STORE THE PC THAT SHOULD
2101                                     ;BE PUSHED ON THE STACK
2102                                     ;IF A PARITY ABORT OCCURS
2103 005420 026000          CMP      -2(RO), RO ;DO A DATI
2104 005424 042777 000004 174170          BIC      #BIT2, @PARITY ;WRITE NORMAL FOR EMT CALL
2105 005432 104001          HLT      +1          ;DIDN'T ABORT
2106 005434 000410          BR      .+22          ;GO TO NEXT TEST
2107 005436 042777 000005 174156 C4:  BIC      #BIT2!BIT0, @PARITY ;WRITE NORMAL AND DISABLE
2108 005444 004037 011550          JSR      RO, @#CHECKLOC ;CHECK FOR GOOD ABORT
2109 005450 104003          HLT      +3          ;ABORTED INCORRECTLY
2110 005452 012706 001100          MOV      #STACK, SP ;RESET THE STACK
2111                                     ;*****
2112                                     ;TEST 30          TEST (ADDRESS) SM7, DMO CMP INSTRUCTION
2113                                     ;*****
2114 005456 000004          †ST30: SCOPE
2115                                     ;          11/45 **** ROM STATE 142 ****
2116                                     ;
2117                                     ;          11/40 **** ROM STATE 245 ****

```

Z


```

2118 005460 004015 JSR RO,(R5) ;SET UP PARITY VECTOR SERVICE
2119 005462 005514 C5 ;ROUTINE ADDRESS
2120 005464 011100 MOV @R1,RO ;SET UP FOR DATO
Z 2121 005466 010020 MOV RO,(RO)+ ;DO THE DATO
2122 005470 012737 005502 001332 MOV #.+12,@#SGDDAT ;STORE THE PC THAT SHOULD
2123 ;BE PUSHED ON THE STACK
2124 ;IF A PARITY ABORT OCCURS
2125 005476 027000 177776 CMP @-2(RO),RO ;DO A DATI
2126 005502 042777 000004 174112 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2127 005510 104001 HLT +1 ;DIDN'T ABORT
2128 005512 000410 BR .+22 ;GO TO NEXT TEST
2129 005514 042777 000005 174100 C5: BIC #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2130 005522 004037 011550 JSR RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
2131 005526 104003 HLT +3 ;ABORTED INCORRECTLY
2132 005530 012706 001100 MOV #STACK,SP ;RESET THE STACK
2133 ;*****
2134 ;TEST 31 TEST (ADDRESS) SM3,DMO CMP INSTRUCTION
2135 ;*****
2136 005534 000004 TST31: SCOPE
2137 ; 11/45 **** ROM STATE 27 ****
2138 ;
2139 ; 11/40 **** ROM STATE 245 ****
2140 005536 004015 JSR RO,(R5) ;SET UP PARITY VECTOR SERVICE
2141 005540 005570 C6 ;ROUTINE ADDRESS
2142 005542 011100 MOV @R1,RO ;SET UP FOR A DATO
2143 005544 010010 MOV RO,@RO ;DO THE DATO
2144 005546 012737 005556 001332 MOV #.+10,@#SGDDAT ;STORE THE PC THAT SHOULD
2145 ;BE PUSHED ON THE STACK
2146 ;IF A PARITY ABORT OCCURS
2147 005554 023000 CMP @ (RO)+,RO ;DO A DATI
2148 005556 042777 000004 174036 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2149 005564 104001 HLT +1 ;DIDN'T ABORT
2150 005566 000410 BR .+22 ;GO TO NEXT TEST
2151 005570 042777 000005 174024 C6: BIC #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2152 005576 004037 011550 JSR RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
2153 005602 104003 HLT +3 ;ABORTED INCORRECTLY
2154 005604 012706 001100 MOV #STACK,SP ;RESET THE STACK
2155 ;*****
2156 ;TEST 32 TEST (ADDRESS) SM5,DMO CMP INSTRUCTION
2157 ;*****
2158 005610 000004 TST32: SCOPE
2159 ; 11/45 **** ROM STATE 27 ****
2160 ;
2161 ; 11/40 **** ROM STATE 245 ****
2162 005612 004015 JSR RO,(R5) ;SET UP PARITY VECTOR SERVICE
2163 005614 005644 C7 ;ROUTINE ADDRESS
2164 005616 011100 MOV @R1,RO ;SET UP FOR A DATO
Z 2165 005620 010020 MOV RO,(RO)+ ;DO THE DATO
2166 005622 012737 005632 001332 MOV #.+10,@#SGDDAT ;STORE THE PC THAT SHOULD
2167 ;BE PUSHED ON THE STACK
2168 ;IF A PARITY ABORT OCCURS
2169 005630 025000 CMP @-(RO),RO ;DO A DATI
2170 005632 042777 000004 173762 BIC #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2171 005640 104001 HLT +1 ;DIDN'T ABORT
2172 005642 000410 BR .+22 ;GO TO NEXT TEST
2173 005644 042777 000005 173750 C7: BIC #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE

```

BOS

MAINDEC-11-DCABR-E
DCABRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 50
CPU PARITY TEST MAIN FLOW

```

2174 005652 004037 011550      JSR    RO, @CHECKLOC      :CHECK FOR GOOD ABORT
2175 005656 104003              HLT    +3                  :ABORTED INCORRECTLY
2176 005660 012706 001100      MOV    @STACK, SP        :RESET THE STACK
*****
2178      :TEST 33      TEST (DATA) SM7, DMO CMP INSTRUCTION
*****
2180 005654 000004      TST33: SCOPE
                11/45 **** ROM STATE 146 ****
:
                11/40 **** ROM STATE 250 ****
2184 005666 004015      JSR    RO, (R5)           :SET UP PARITY VECTOR SERVICE
2185 005670 005742      CB                      :ROUTINE ADDRESS
2186 005672 011100      MOV    @R1, RO           :SET UP FOR A DATO
2187 005674 042777 000004 173720      BIC    @BIT2, @PARITY    :WRITE NORMAL
2188 005702 010060 177776      MOV    RO, -2(R0)       :DO A DATO (ADDRESS NORMAL)
2189 005706 052777 000004 173706      BIS    @BIT2, @PARITY    :WRITE OTHER PARITY
2190 005714 011110      MOV    @R1, @RO         :DO A DATO (DATA OTHER PARITY)
2191 005716 012737 005730 001332      MOV    @.+12, @SGDDAT    :STORE THE PC THAT SHOULD
                :BE PUSHED ON THE STACK
                :IF A PARITY ABORT OCCURS
2194 005724 027000 177776      CMP    @-2(R0), RO      :DO A DATI
2195 005730 042777 000004 173664      BIC    @BIT2, @PARITY    :WRITE NORMAL FOR EMT CALL
2196 005736 104001      HLT    +1                :DIDN'T ABORT
2197 005740 000410      BR     .+22              :GO TO NEXT TEST
2198 005742 042777 000005 173652 CB:      BIC    @BIT2!BIT0, @PARITY :WRITE NORMAL AND DISABLE
2199 005750 004037 011550      JSR    RO, @CHECKLOC    :CHECK FOR GOOD ABORT
2200 005754 104003              HLT    +3                  :ABORTED INCORRECTLY
2201 005756 012706 001100      MOV    @STACK, SP        :RESET THE STACK
*****
2203      :TEST 34      TEST DM3 JMP INSTRUCTION
*****
2205 005762 000004      TST34: SCOPE
                11/45 **** ROM STATE 221 ****
:
                11/40 **** ROM STATE 303 ****
2209 005764 004015      JSR    RO, (R5)           :SET UP PARITY VECTOR SERVICE
2210 005766 006020      D                      :ROUTINE ADDRESS
2211 005770 011100      MOV    @R1, RO -        :SET UP FOR A DATO
2212 005772 012710 006006      MOV    @DD, @RO         :DO THE DATO
2213 005776 012737 006006 001332      MOV    @.+10, @SGDDAT    :STORE THE PC THAT SHOULD
                :BE PUSHED ON THE STACK
                :IF A PARITY ABORT OCCURS
2216 006004 000130      JMP    @R0+              :DO A DATI
2217 006006 042777 000004 173606 DD:      BIC    @BIT2, @PARITY    :WRITE NORMAL FOR EMT CAL
2218 006014 104001      HLT    +1                :DIDN'T ABORT
2219 006016 000410      BR     .+22              :GO TO NEXT TEST
2220 006020 042777 000005 173574 D:      BIC    @BIT2!BIT0, @PARITY :WRITE NORMAL AND DI;ABLE
2221 006026 004037 011550      JSR    RO, @CHECKLOC    :CHECK FOR GOOD ABORT
2222 006032 104003              HLT    +3                  :ABORTED INCORRECTLY
2223 006034 012706 001100      MOV    @STACK, SP        :RESET THE STACK
*****
2225      :TEST 35      TEST DM5 JMP INSTRUCTION
*****
2227 006040 000004      TST35: SCOPE
                11/45 **** ROM STATE 231 ****
:

```

```

2230      :      11/40 **** ROM STATE 303 ****
2231 006042 004015      JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
2232 006044 006076      DO          ;ROUTINE ADDRESS
2233 006046 011100      MOV      @R1,RO      ;SET UP FOR A DATO
2234 006050 012720 006064  MOV      @DD0,(RO)+    ;DO THE DATO
2235 006054 012737 006064 001332  MOV      @.+10,@$GDDAT ;STORE THE PC THAT SHOULD
2236      :          ;BE PUSHED ON THE STACK
2237      :          ;IF A PARITY ABORT OCCURS
2238 006062 000150      JMP      @-(RO)      ;DO A DATI
2239 006064 042777 000004 173530 DD0:  BIC      @BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2240 006072 104001      HLT      +1          ;DIDN'T ABORT
2241 006074 000410      BR       .+22        ;GO TO NEXT TEST
2242 006076 042777 000005 173516 DD:  BIC      @BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2243 006104 004037 011550      JSR      RO,@CHECKLOC ;CHECK FOR GOOD ABORT
2244 006110 104003      HLT      +3          ;ABORTED INCORRECTLY
2245 006112 012706 001100      MOV      @STACK,SP   ;RESET THE STACK
2246      :          ;*****
2247      :          ;TEST 36      TEST DM7 JMP INSTRUCTION
2248      :          ;*****
2249 006116 000004      ;TEST36: SCOPE
2250      :          ;          11/45 **** ROM STATE 231 ****
2251      :          ;
2252      :          ;
2253      :          ;          11/40 **** ROM STATE 303 ****
2254 006120 004015      JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
2255 006122 006156      DI          ;ROUTINE ADDRESS
2256 006124 011100      MOV      @R1,RO      ;SET UP FOR A DATO
2257 006126 012720 006144  MOV      @DD1,(RO)+    ;DO THE DATO
2258 006132 012737 006144 001332  MOV      @.+12,@$GDDAT ;STORE THE PC THAT SHOULD
2259      :          ;BE PUSHED ON THE STACK
2260      :          ;IF A PARITY ABORT OCCURS
2261 006140 000170 177776      JMP      @-2(RO)     ;DO A DATI
2262 006144 042777 000004 173450 DD1:  BIC      @BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2263 006152 104001      HLT      +1          ;DIDN'T ABORT
2264 006154 000410      BR       .+22        ;GO TO NEXT TEST
2265 006156 042777 000005 173436 DD:  BIC      @BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2266 006164 004037 011550      JSR      RO,@CHECKLOC ;CHECK FOR GOOD ABORT
2267 006170 104003      HLT      +3          ;ABORTED INCORRECTLY
2268 006172 012706 001100      MOV      @STACK,SP   ;RESET THE STACK
2269

```

```

:*****
:*****
:*****
:*****

```

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

1ST PUSH - ADDRESS OF THE TAG 'VECSET' (NORMAL)
THIS ADDRESS WOULD BE PLACED IN
R5 UPON COMPLETION OF 'RTS R5'
INSTRUCTION

2ND PUSH

:
:
: NO. OF PARAMETERS AS A FUNCTION
: OF MEMORY INTERLEAVING


```

2342 006230 001404      BEQ      2$
2343 006232 005302      DEC      R2
2344 006234 012746 000001  MOV     #1,-(SP)
2345 006240 000772      BR      1$
2346 006242 052777 000004 173352 2$:  BIS     #BIT2,%PARITY
2347 006250 017702 173172  MOV     %INTERAD,R2
2348
2349 006254 005302      DEC      R2
2350
2351
2352 006256 062702 006400  ADD     #6400,R2
2353
2354 006262 010246  MOV     R2,-(SP)
2355 006264 042777 000004 173330  BIC     #BIT2,%PARITY
2356 006272 010605  MOV     SP,R5
2357
2358 006274 010602  MOV     SP,R2
2359
2360 006276 005737 001642  TST     %CPU40
2361 006302 001002      BNE     3$
2362
2363
2364
2365 006304 062702 000002  ADD     #2,R2
2366
2367
2368 006310 010237 001332      3$:  MOV     R2,%$GDDAT
2369
2370
2371 006314 004767 000004      JSR     PC,MRKO
2372 006320 104001      E:  HLT     +1
2373 006322 000407      BR      .+20
2374 006324 000205  MRKO:  RTS     R5
2375 006326 042777 000001 173266  E0:  BIC     #BIT0,%PARITY
2376 006334 004037 011550      JSR     R0,%CHECKLOC
2377 006340 104003      HLT     +3
2378 006342 013706 001476  MOV     %NEWSTK,SP
2379
2380
2381

```

```

: BRANCH IF NO
: SUBTRACT 1 FROM PARAMETER COUNT
: PUSH PARAMETER ONTO STACK
: GO BACK TO SEE IF ANY MORE!
: WRITE OTHER PARITY
: GET THE INTERLEAVE FACTOR
: FOR THIS CONTROLLER
: CALCULATE NO. OF PARAMETERS
: THAT WERE TO BE PUSHED ON THE
: STACK
: CALCULATE THE CORRESPONDING
: MARK INSTRUCTION
: PUSH MARK INSTRUCTION ON STACK
: WRITE NORMAL
: PLACE MARK INSTRUCTION ADDRESS
: INTO R5 FOR RTS
: GET THE PC OF THE MARK
: INSTRUCTION
: ARE WE ON AN 11/40?
: BRANCH IF YES
: AND DON'T STEP UP THE PC
: IT WON'T BE UPDATED ON THE
: PARITY ABORT
: STEP UP THE PC
: IT WILL BE UPDATED ON THE
: PARITY ABORT
: STORE THIS VALUE OF PC
: THAT SHOULD BE PUSHED ON THE
: STACK IF A PARITY ABORT OCCURS
: SUBROUTINE CALL
: DIDN'T ABORT
: GO TO RESET THE STACK
: RETURN FROM SUBROUTINE
: DISABLE PARITY
: CHECK FOR GOOD ABORT
: ABORTED INCORRECTLY
: RESET 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

:NTH PUSH
:NTH +1 PUSH - MARK INSTRUCTION (NORMAL)
:LAST PUSH - OLD PC FROM THE 'JSR' (NORMAL)

:NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
RS CONTENTS (1ST PUSH)

WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
PROPERLY UPDATED, THUS GIVING,

:1ST PUSH - PS FROM THE PARITY ERROR
:2ND PUSH - PC FROM THE PARITY ERROR

2416										
2417										
2418										
2419	006346	012705	011450			MOV	#VECSET,R5			:RESTORE THE PARITY VECTOR
2420										:SERVICE ADDRESS SETUP ROUTINE
2421										:ADDRESS
2422										*****
2423						:TEST 40	TEST	OLD REGISTER CONTENTS	MARK INSTRUCTION	*****
2424										*****
2425	006352	000004				1ST40: SCOPE				
2426								11/45 **** ROM STATE 235 ****		
2427										
2428								11/40 **** ROM STATE 356 ****		
2429	006354	004015				JSR	RO,(R5)			:SET UP PARITY VECTOR SERVICE
2430	006356	006510				E2				:ROUTINE ADDRESS
2431	006360	017702	173062			MOV	2#INTERAD,R2			:GET THE INTERLEAVE FACTOR
2432										:FOR THIS CONTROLLER
2433	006364	005302				DEC	R2			:CALCULATE NO. OF PARAMETERS
2434										:TO BE PUSHED ON THE STACK
2435	006366	042777	000004	173226		BIC	#BIT2,2#PARITY			:WRITE NORMAL
2436	006374	005737	001642			TST	2#CPU40			:ARE WE ON AN 11/40?
2437	006400	001407				BEQ	3\$:BRANCH IF NO
2438										:DATA WILL BE TAKEN FIRST AND
2439										:THEN THE PC UPDATED
2440	006402	013700	001476			MOV	2#NEWSTK,RO			:GET THE INITIAL STACK PCINT
2441	006406	162700	000002			SUB	#2,RO			:DROP DOWN 1 WORD ADDRESS
2442										:BECAUSE THE PC WILL BE UPDATED
2443										:FIRST THEN THE DATA TAKEN
2444	006412	010037	001332			MOV	RO,2#SGDDAT			:STORE THIS VALUE AS THE PC
2445										:THAT SHOULD BE PUSHED ON THE STACK
2446										:IF A PARITY ABORT OCCURS
2447	006416	000403				BR	4\$:CONTINUE WITH TEST
2448	006420	012737	006502	001332	3\$:	MOV	#E1,2#SGDDAT			:STORE THE PC THAT SHOULD
2449										:BE PUSHED ON THE STACK
2450										:IF A PARITY ABORT OCCURS
2451	006426	052777	000004	173166	4\$:	BIS	#BIT2,2#PARITY			:WRITE OTHER PARITY
2452	006434	010546				MOV	RS,-(SP)			:STORE OLD RS CONTENTS ON STACK
2453	006436	042777	000004	173156		BIC	#BIT2,2#PARITY			:WRITE NORMAL


```

2510 ; COMPLETION OF SOB INSTRUCTION
2511 ; THE INSTRUCTION WILL BE IN
2512 ; PARITY MEMORY AS DESIGNATED BY
2513 ; THE CONTENTS OF R1
2514 006620 042777 000004 172774 BIC #BIT2,PARITY
2515 006626 012712 000203 . MOV #203,R2
2516 ; WRITE NORMAL
2517 ; MOVE AN RTS R3 INTO THE LOCATION
2518 ; FOLLOWING THE INSTRUCTION TO BE
2519 ; EXECUTED IN PARITY MEMORY IN
2520 006632 000403 BR .+10
2521 006634 004342 F: JSR R3,-(R2)
2522 ; GO TO SOB INSTRUCTION
2523 ; THE INSTRUCTION TO BE DONE ON
2524 ; COMPLETION OF SOB EXECUTION IS
2525 ; PLACED HERE
2526 006636 104001 HLT +1
2527 006640 000407 BR .+20
2528 006642 077004 SOB RO,F
2529 006644 042777 000001 172750 FO: BIC #BIT0,PARITY
2530 006652 004037 011550 JSR RO,#CHECKLOC
2531 006656 104003 HLT +3
2532 006660 013706 001476 MOV #NEWSTK,SP
2533 ; *****
2534 ; TEST 42 TEST SMO,DMD MOV INSTRUCTION
2535 ; *****
2536 ST42: SCOPE
2537 ;
2538 ; 11/45 **** ROM STATE 20 ****
2539 ;
2540 ;
2541 ; 11/40 **** ROM STATE 1 ****
2542 ;
2543 JSR RO,(R5) ; SET UP PARITY VECTOR SERVICE
2544 FI ; ROUTINE ADDRESS
2545 MOV #R1,RO ; SET UP FOR A DATO
2546 MOV #INTERAD,R2 ; GET THE INTERLEAVE FACTOR
2547 ; FOR THIS CONTROLLER
2548 006700 042777 000004 172714 15: BIC #BIT2,PARITY
2549 006706 005702 TST R2 ; WRITE NORMAL
2550 ; ANY 'MOV RO,RO'S TO BE SHOVED
2551 ; INTO PARITY MEMORY AREA?
2552 006710 001404 BEQ 25 ; BRANCH IF NO
2553 006712 005302 DEC R2 ; SUBTRACT 1 FROM INSTRUCTION
2554 ; COUNT
2555 006714 012720 010000 MOV #010000,(RO)+ ; MOVE THE INSTRUCTION 'MOV RO,RO'
2556 ; TO PARITY AREA
2557 006720 000772 BR 15 ; GO BACK TO SEE IF ANY MORE!
2558 006722 052777 000004 172672 25: BIS #BIT2,PARITY
2559 006730 012720 010000 MOV #010000,(RO)+ ; WRITE OTHER PARITY
2560 ; MOVE THE INSTRUCTION 'MOV RO,RO'
2561 ; INTO THE NEXT WORD LOCATION AFTER
2562 ; THE PREVIOUS 'MOV RO,RO' INSTRUCTION
2563 006734 042777 000004 172660 BIC #BIT2,PARITY
2564 006742 012710 000203 MOV #203,(RO)
2565 ; WRITE NORMAL
2566 ; MOVE 'RTS R3' INTO NEXT WORD
2567 ; LOCATION AFTER THE PREVIOUS
2568 ; 'MOV RO,RO' INSTRUCTION IN CASE
2569 ; A PARITY ABORT DOESN'T OCCUR
2570 006746 005737 001642 TST #CPU40
2571 006752 001402 BEQ 35 ; ARE WE ON AN 11/40?
2572 ; BRANCH IF NO
2573 ; SINCE PC WILL BE UPDATED ON THE
2574 ; PARITY ABORT
2575 006754 162700 000002 SUB #2,RO ; DROP BACK 1 WORD ADDRESS
2576 ; SINCE THE PC WILL NOT BE

```



```

2566                                     ;UPDATED ON THE PARITY ABORT
2567 006760 010037 001332          3S:  MOV      RO,2#SGDDAT      ;STORE THE PC THAT SHOULD
2568                                     ;BE PUSHED ON THE STACK
2569                                     ;IF A PARITY ABORT OCCURS
2570 006764 004371 000000          JSR      R3,2(R1)      ;GO TO PARITY MEMORY AREA
2571 006770 104001                   HLT      +1           ;DIDN'T ABORT
2572 006772 000406                   BR       .+16        ;GO TO RESET THE STACK
2573 006774 042777 000001 172620 F1: BIC      #BIT0,2#PARITY ;DISABLE PARITY
2574 007002 004037 011550          JSR      RO,2#CHECKLOC ;CHECK FOR GOOD ABORT
2575 007006 104003                   HLT      +3           ;ABORTED INCORRECTLY
2576 007010 013706 001476          MOV      2#NEWSTK,SP ;RESET THE STACK
2577
2578
2579

```

```

*****
*****
*****
*****

```

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

1ST PUSH - OLD R3 FROM THE 'JSR' (OTHER PARITY)

NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE R3 CONTENTS (1ST PUSH)

```

*****
*****
*****
*****

```

```

2598
2599
2600                                     ;*****
2601 TEST 43          TEST  RTS INSTRUCTION ;*****
2602                                     ;*****
2603 007014 000004          ST43: SCOPE          ;*****
2604                                     ;*****
2605                                     ;*****
2606                                     ;*****
2607                                     ;*****
2608                                     ;*****
2609 007016 004015          JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
2610 007020 007066          GO          ;ROUTINE ADDRESS
2611 007022 042777 000004 172572      BIC      #BIT2,2#PARITY ;WRITE NORMAL
2612 007030 012700 007056          MOV      #G,RO      ;SET UP A SUBROUTINE ADDRESS
2613 007034 012737 007052 001332      MOV      #.+16,2#SGDDAT ;STORE THE PC THAT SHOULD
2614                                     ;BE PUSHED ON THE STACK
2615 007042 052777 000004 172552      BIS      #BIT2,2#PARITY ;IF A PARITY ABORT OCCURS
2616 007050 004320          JSR      R3,(RO)+    ;WRITE OTHER PARITY
2617 007052 104001          HLT      +1           ;SUBROUTINE CALL
2618 007054 000412          BR       .+26        ;DIDN'T ABORT
2619 007056 042777 000004 172536 G:   BIC      #BIT2,2#PARITY ;GO TO RESET THE STACK
2620 007064 000203          RTS      R3         ;WRITE NORMAL
2621 007066 042777 000001 172526 GO:  BIC      #BIT0,2#PARITY ;RETURN FROM SUBROUTINE
2622 007074 004037 011550          JSR      RO,2#CHECKLOC ;DISABLE PARITY
2623                                     ;CHECK FOR GOOD ABORT

```

Z

```

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

```

```

:*****
:*****
:*****
:*****

```

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

```

:1ST PUSH - OLD PS FOR 'RTI' (OTHER PARITY)
:2ND PUSH - OLD PC FOR 'RTI' (OTHER PARITY)

```

NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
THE OLD PC (2ND PUSH)

WHEN THE PARITY ERROR OCCURS THE STACK POINTER
IS POSITIONED AT THE 1ST PUSH, THUS GIVING,
THE 'NEW' PS FOR THE RTI INSTRUCTION WHICH
WILL OVERLAY THE OLD PC SET UP FOR THE 'RTI'
INSTRUCTION (2ND PUSH) THUS GIVING,

```

:1ST PUSH - OLD PS FOR 'RTI' (OTHER PARITY)
:2ND PUSH - NEW PS FROM THE 'RTI'
:3RD PUSH - NEW PC FROM THE 'RTI'

```

```

:*****
:*****
:*****
:*****

```

```

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

```

```

:*****
:TEST 44          TEST 1ST POP RTI INSTRUCTION
:*****
TST44: SCOPE
:
:          11/45 **** ROM STATE 212 ****
:
:          11/40 **** ROM STATE 320 ****

```

```

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

```

```

:*****
:*****
:*****
:*****

```

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

- 1ST PUSH - OLD PS FOR 'RTI' (OTHER PARITY)
- 2ND PUSH - OLD PC FOR 'RTI' (OTHER PARITY)

NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
THE OLD PS (1ST PUSH)

THE 2ND PUSH IS REWRITTEN NORMAL BEFORE
DOING THE 'RTI'

WHEN THE PARITY ERROR OCCURS THE STACK
POINTER IS PROPERLY UPDATED AND THE NEW
PS AND PC FROM THE RTI INSTRUCTION IS
PUSHED ONTO THE STACK, THUS GIVING,

- 1ST PUSH - NEW PS FROM 'RTI' (NORMAL)
- 2ND PUSH - NEW PC FROM 'RTI' (NORMAL)

```

:*****
:*****
:*****
:*****

```

```

2714
2715
2716
2717
2718
2719 007170 000004
2720
2721
2722
2723 007172 004015
2724 007174 007242
2725 007176 012746 000340
2726 007202 012746 007216
2727 007206 012737 007216 001332
2728
2729
2730 007214 000402
2731 007216 104001
2732 007220 000416
2733 007222 042777 000005 172372

:*****
:TEST 45          TEST 2ND POP RTI INSTRUCTION
:*****
:ST45: SCOPE
:
:          11/45 **** ROM STATE 214 ****
:
:          11/40 **** ROM STATE 322 ****
:
:          JSR      RO, (R5)      ;SET UP PARITY VECTOR SERVICE
:          H2          ;ROUTINE ADDRESS
:          MOV      #340, -(SP)   ;SET PS FOR 'RTI'
:          MOV      #H3, -(SP)   ;SET RETURN FROM 'RTI'
:          MOV      #H3, @#SGDDAT ;STORE THE PC THAT SHOULD
:                                ;BE PUSHED ON THE STACK
:                                ;IF A PARITY ABORT OCCURS
:          BR       H4          ;GO DO 'RTI'
:          HLT      +1          ;DIDN'T ABORT
:          BR       .+36        ;GO TO RESET THE STACK
:          BIC      #BIT2!BIT0, @PARITY ;WRITE NORMAL AND DISABLE PARITY

```

```

2734 007230 012646          MOV      (SP)+,-(SP)          ;WRITE 1ST 'POP' NORMAL SO NO
2735                                ;ERROR WILL OCCUR
2736 007232 052777 000001 172362  BIS      #BIT0,@PARITY      ;ENABLE PARITY
2737 007240 000002          RTI                          ;RETURN FROM INTERRUPT
2738 007242 042777 000001 172352 H2:  BIC      #BIT0,@PARITY      ;DISABLE PARITY
2739 007250 004037 011550          JSR      RO,@#CHECKLOC      ;CHECK FOR GOOD ABCRT
2740 007254 104003          HLT      +3                  ;ABORTED INCORRECTLY
2741 007256 012706 001100          MOV      #STACK,SP          ;INITIALIZE THE STACK BACK
2742                                ;TO 1100 FOR WE DON'T NEED IT
2743                                ;TO BE IN PARITY MEMORY AREA
2744                                ;ANYMORE
2745                                ;*****
2746                                ;TEST 46      TEST (DATA) SM6,DM0  ADD INSTRUCTION
2747                                ;*****
2748 007262 000004          TST46: SCOPE
2749                                ;
2750                                ;
2751                                ;
2752                                ;
2753                                ;
2754                                ;
2755                                ;
2756                                ;
2757                                ;
2758                                ;
2759                                ;
2760                                ;
2761                                ;
2762                                ;
2763                                ;
2764                                ;
2765                                ;
2766                                ;
2767                                ;
2768                                ;
2769                                ;
2770                                ;
2771                                ;
2772                                ;
2773                                ;
2774                                ;
2775                                ;
2776                                ;
2777                                ;
2778                                ;
2779                                ;
2780                                ;
2781                                ;
2782                                ;
2783                                ;
2784                                ;
2785                                ;
2786                                ;
2787                                ;
2788                                ;
2789                                ;

```

Z

TEST 46 TEST (DATA) SM6,DM0 ADD INSTRUCTION

TST46: SCOPE
11/45 **** ROM STATE 142 ****

11/40 **** ROM STATE 250 ****

;SET UP PARITY VECTOR SERVICE
;ROUTINE ADDRESS
;SET UP FOR A DATO
;DO THE DATO
;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
;IF A PARITY ABORT OCCURS
;DO A DATI
;WRITE NORMAL FOR EMT CALL
;DIDN'T ABORT
;GO TO NEXT TEST
;WRITE NORMAL AND DISABLE
;CHECK FOR GOOD ABORT
;ABORTED INCORRECTLY
;RESET THE STACK

TEST 47 TEST DM1 TSTB INSTRUCTION

TST47: SCOPE
11/45 **** ROM STATE 175 ****

11/40 **** ROM STATE 267 ****

;SET UP PARITY VECTOR SERVICE
;ROUTINE ADDRESS
;SET UP FOR A DATO
;DO THE DATO
;STORE THE PC THAT SHOULD
;BE PUSHED ON THE STACK
;IF A PARITY ABORT OCCURS
;DO A DATI, DATIP
;WRITE NORMAL FOR EMT CALL
;DIDN'T ABORT
;GO TO NEXT TEST
;WRITE NORMAL AND DISABLE
;CHECK FOR GOOD ABORT
;ABORTED INCORRECTLY
;RESET THE STACK

M05

MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 61
CPU PARITY TEST MAIN FLOW

```

2790      :TEST 50      TEST DM3 CLRB INSTRUCTION
2791      :*****
2792 007414 000004  †ST50: SCOPE
2793      :                11/45 **** ROM STATE 221 ****
2794      :
2795      :                11/40 **** ROM STATE 264 ****
2796 007416 004015      JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
2797 007420 007450      NN      ;ROUTINE ADDRESS
2798 007422 011100      MOV      @R1,RO      ;SET UP FOR A DATO
2799 007424 010010      MOV      RO,(RO)     ;DO THE DATO
2800 007426 012737 007436 001332      MOV      #.+10,@#$GDDAT ;STORE THE PC THAT SHOULD
2801      :                ;BE PUSHED ON THE STACK
2802      :                ;IF A PARITY ABORT OCCURS
2803 007434 105030      CLRB     @ (RO)+     ;DO A DATI
2804 007436 042777 000004 172156      BIC      #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2805 007444 104001      HLT     +1          ;DIDN'T ABORT
2806 007446 000410      BR      .+22        ;GO TO NEXT TEST
2807 007450 042777 000005 172144 NN:      BIC      #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2808 007456 004037 011550      JSR      RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
2809 007462 104003      HLT     +3          ;ABORTED INCORRECTLY
2810 007464 012706 001100      MOV      #STACK,SP   ;RESET THE STACK
2811      :*****
2812      :TEST 51      TEST SM1 SUBTRACT INSTRUCTION
2813      :*****
2814 007470 000004  †ST51: SCOPE
2815      :                11/45 **** ROM STATE 27 ****
2816      :
2817      :                11/40 **** ROM STATE 250 ****
2818 007472 004015      JSR      RO,(R5)     ;SET UP PARITY VECTOR SERVICE
2819 007474 007526      P      ;ROUTINE ADDRESS
2820 007476 011100      MOV      @R1,RO     ;SET UP FOR A DATO
2821 007500 010010      MOV      RO,(RO)   ;DO THE DATO
2822 007502 012737 007512 001332      MOV      #.+10,@#$GDDAT ;STORE THE PC THAT SHOULD
2823      :                ;BE PUSHED ON THE STACK
2824      :                ;IF A PARITY ABORT OCCURS
2825 007510 161027 177777      SUB      (RO),#177777 ;DO A DATI
2826 007514 042777 000004 172100      BIC      #BIT2,@PARITY ;WRITE NORMAL FOR EMT CALL
2827 007522 104001      HLT     +1          ;DIDN'T ABORT
2828 007524 000410      BR      .+22        ;GO TO NEXT TEST
2829 007526 042777 000005 172066 P:      BIC      #BIT2!BIT0,@PARITY ;WRITE NORMAL AND DISABLE
2830 007534 004037 011550      JSR      RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
2831 007540 104003      HLT     +3          ;ABORTED INCORRECTLY
2832 007542 012706 001100      MOV      #STACK,SP   ;RESET THE STACK
2833      :*****
2834      :TEST 52      TEST (DATA) SMS BISB INSTRUCTION
2835      :*****
2836 007546 000004  †ST52: SCOPE
2837      :                11/45 **** ROM STATE 177 ****
2838      :
2839      :                11/40 **** ROM STATE 245 ****
2840 007550 004015      JSR      RO,(R5)     ;SET UP PARITY VECTOR SERVICE
2841 007552 007604      R      ;ROUTINE ADDRESS
2842 007554 011100      MOV      @R1,RO     ;SET UP FOR A DATO
2843 007556 010010      MOV      RO,(RO)   ;DO THE DATO
2844 007560 012737 007570 001332      MOV      #.+10,@#$GDDAT ;STORE THE PC THAT SHOULD
2845      :                ;BE PUSHED ON THE STACK

```

```

2846                                     ; IF A PARITY ABORT OCCURS
2847 007566 155060 000002             BISB  2-(RO),2(RO)           ; DO A DATI
2848 007572 042777 000004 172022     BIC   #BIT2,2PARITY       ; WRITE NORMAL FOR EMT CALL
2849 007600 104001                   HLT   +1                     ; DIDN'T ABORT
2850 007602 000410                   BR    +22                    ; GO TO NEXT TEST
2851 007604 042777 000005 172010 R:   BIC   #BIT2:BIT0,2PARITY   ; WRITE NORMAL AND DISABLE
2852 007612 004037 011550             JSR   RO,2#CHECKLOC       ; CHECK FOR GOOD ABORT
2853 007616 104003                   HLT   +3                     ; ABORTED INCORRECTLY
2854 007620 012706 001100             MOV   #STACK,SP          ; RESET THE STACK
2855                                     ;*****
2856                                     ;TEST 53             NEXT 4 TESTS ARE NON-INTERLEAVE 4K DEPENDENT
2857                                     ;*****
2858 007624 000004                   TST53: SCOPE
2859 007626 022701 017360             CMP   #17360,R1          ; HAVE WE PARITY IN THE LOWER 4K?
2860                                     ; AND IS THE SELECTED REGISTER
2861                                     ; GOVERNING THE 4K AREA?
2862                                     ; THIS COMPARE IS IF THE KT11
2863                                     ; ISN'T ENABLED DURING PROGRAM
2864                                     ; EXECUTION
2865 007632 101005                   BHI   2$                    ; YES - PROCEED TO NEXT TESTS
2866                                     ; WHICH DEPEND ON PARITY IN LOWER 4K
2867 007634 022777 000140 171554     CMP   #140,2$SETAD       ; THE ABOVE COMPARE DIDN'T
2868                                     ; CHECK - - DO THIS COMPARE TO
2869                                     ; SEE IF IT WAS BECAUSE THE
2870                                     ; TABLE WAS CREATED WITH MEMORY
2871                                     ; MANAGEMENT TURNED ON
2872 007642 001401                   BEQ   2$                    ; PROCEED TO NEXT TESTS IF THIS
2873                                     ; COMPARE CHECKS
2874 007644 000404                   BR    4$                    ; THE REGISTER UNDER TEST IS NOT
2875                                     ; CONTROLLING THE LOWER 4K!!!
2876                                     ; GO TO RE-EVALUATE $STNM AND
2877                                     ; JUMP OVER THE (4) 4K DEPENDENT
2878                                     ; TESTS
2879 007646 022777 000001 171572 2$:   CMP   #1,2$INTERAD       ; IS THIS CONTROLLER
2880                                     ; INTERLEAVED??
2881 007654 001405                   BEQ   3$                    ; BRANCH IF NO
2882 007656 062767 000004 171416 4$:   ADD   #4,$STNM           ; SET $STNM TO PROPER VALUE
2883                                     ; SINCE THE NEXT 4 TESTS WILL
2884                                     ; BE SKIPPED
2885 007664 000137 010426             JMP   2#RED+26            ; JUMP TO 1ST INDEX WORD TEST
2886 007670 005237 001624             3$:   INC   2#PSPCORZONES   ; SET FLAG INDICATING TO 'CHECKLOC'
2887                                     ; ROUTINE THAT PS AND PC FETCH
2888                                     ; AND RED & YELLOW ZONE AREAS
2889                                     ; ARE GOING TO BE TESTED
2890
2891
2892

```

```

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

```

```

;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)

```

```

:2ND PUSH - OLD PC FROM ERROR TRAP (NORMAL)
:NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PS
      WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
      ALTERED FROM THE ORIGINAL ERROR TRAP, THUS GIVING.
:1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
:2ND PUSH - NEW PS FROM THE PARITY ERROR
:3RD PUSH - NEW PC FROM THE PARITY ERROR

```

```

*****
*****
*****
*****

```

2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948

```

*****
:TEST 54 TEST NEW 'PS' FETCH
*****
:STS4: SCOPE
      11/45 **** ROM STATE 357 ****
:
      11/40 **** ROM STATE 115 ****
      JSR RO,(RS) ;SET UP PARITY VECTOR SERVICE
      TO ;ROUTINE ADDRESS
      BIC #BIT2,PARITY ;WRITE NORMAL
      MOV #T,RESVEC ;RESERVED INSTRUCTION TIMEOUT
      ;VECTOR ADDRESS
      MOV #T,SGDDAT ;STORE THE PC THAT SHOULD
      ;BE PUSHED ON THE STACK
      ;IF A PARITY ABORT OCCURS
      BIS #BIT2,PARITY ;WRITE OTHER PARITY
      MOV #340,RESVEC+2 ;NEW 'PS' FOR ERROR TRAP
      BIC #BIT2,PARITY ;WRITE NORMAL
      7000 ;NON-RECOGNIZABLE OP-CODE
      ;SHOULD ATTEMPT TO TRAP TO 'T:'
      T: HLT +1 ;DIDN'T ABORT
      BR .+16 ;GO TO RESET THE STACK
      TO: BIC #BIT0,PARITY ;DISABLE PARITY
      JSR RO,CHECKLOC ;CHECK FOR GOOD ABORT
      HLT +3 ;ABORTED INCORRECTLY
      MOV #STACK,SP ;RESET THE STACK

```

```

*****
*****
*****
*****

```

```

:THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
:1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)

```

```

:2ND PUSH - OLD PC FROM ERROR TRAP (NORMAL)
:NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PC
:
:WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
:NOT ALTERED FROM THE ORIGINAL ERROR TRAP AND THE
:PC FOR THE PARITY ERROR IS THE OLD PS, THUS GIVING,
:
:1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
:2ND PUSH - OLD PS FROM ERROR TRAP (NORMAL)
:
:IN OTHER WORDS THE ORIGINAL ERROR TRAP AND VECTOR IS LOST!!!

```

```

:*****
:*****
:*****
:*****

```

2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012

```

:*****
:TEST 55 TEST NEW 'PC' FETCH
:*****
1ST55: SCOPE
:
: 11/45 **** ROM STATE 355 ****
:
: 11/40 **** ROM STATE 333 ****
:
: JSR RO,(RS) ; SET UP PARITY VECTOR SERVICE
: VO ; ROUTINE ADDRESS
: MOV #V,#RESVEC ; RESERVED INSTRUCTION TIMEOUT
: ; VECTOR ADDRESS
: MOV #340,#RESVEC+2 ; NEW 'PS' FOR ERROR TRAP
: TST #CPU40 ; ARE WE ON AN 11/40?
: BEG 1$ ; BRANCH IF NO
: MOV #V,#SGDDAT ; STORE THIS VALUE OF THE PC THAT
: ; SHOULD BE PUSHED ON THE STACK
: ; IF A PARITY ABORT OCCURS
: ; SINCE THE PC IS UPDATED FIRST
: ; THEN THE VECTOR DATA TAKEN
: ; CONTINUE WITH TEST
: 1$: MOV #10,#SGDDAT ; STORE THE PC THAT SHOULD
: ; BE PUSHED ON THE STACK
: ; IF A PARITY ABORT OCCURS
: 2$: BIC #BIT2,#PARITY ; WRITE NORMAL
: 7000 ; NON-RECOGNIZABLE OP-CODE
: ; SHOULD ATTEMPT TO TRAP TO 'V:'
: ; DIDN'T ABORT
: V: HLT +1 ; GO TO RESET THE STACK
: BR .+16 ; DISABLE PARITY
: VO: BIC #BIT0,#PARITY ; CHECK FOR GOOD ABORT
: JSR RO,#CHECKLOC ; ABORTED INCORRECTLY
: HLT +3 ; RESET THE STACK
: M'V #STACK.SP

```

```

:*****

```


THE FOLLOWING TEST WILL/SHOULD CAUSE A PARITY ABORT IN THE
'YELLOW' ZONE. THE 'YELLOW' ZONE IS AN AREA UP TO A 16 WORD
LOCATION LIMIT BEYOND THE STACK OVERFLOW BOUNDARY OF 400
(OCTAL). I.E. LOCATIONS 376 - 340 COMPRISE THE YELLOW ZONE.

SINCE PARITY ERRORS HAVE HIGHEST PRIORITY WE WILL BE LOOKING
FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
TRAP TO LOCATION 4.

THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
SHOULD BE AS FOLLOWS:

- 1ST PUSH - PS FROM THE PARITY ERROR
- 2ND PUSH - PC FROM THE PARITY ERROR
- 3RD PUSH - PS FROM THE STACK VIOLATION
- 4TH PUSH - PC FROM THE STACK VIOLATION

NOTE: THE ABOVE CONTENTS WILL EXIST ON THE STACK IF BOTH
THE PARITY ERROR AND THE STACK VIOLATION WERE
RECOGNIZED AND THE PARITY ERROR TOOK PRECEDENCE!

3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069

010076 000004

```

:*****
:TEST S6          TEST ABORT IN 'YELLOW' ZONE
:*****
TST56:  SCOPE
:              11/45 **** ROM STATE 177 ****
:
:              11/40 **** ROM STATE 267 ****

```

010100 004015
010102 010162
010104 005037 000340

```

JSR      RD,(R5)
V1
CLR      2#340

```

```

:SET UP PARITY VECTOR SERVICE
:ROUTINE ADDRESS
:CLEAR BOTTOM LIMIT LOCATION
:OF 'YELLOW' ZONE USING 'OTHER'
:PARITY

```

010110 042777 000004 171504
010116 012706 000376
010122 013727 000004

```

BIC      #BIT2,2#PARITY
MOV      #376,SP
MOV      2#4,(PC)+

```

```

:WRITE NORMAL
:SET STACK IN 'YELLOW' AREA
:SAVE CONTENTS OF LOC. 4 IN
:NEXT LOCATION
:ORIGINAL CONTENTS OF LOC. 4
:GO HERE

```

010126 000000

SAVLOC4: .WORD 0

010130 012737 010166 000004

MOV #V2,2#ERRVEC

```

:SET UP A TIMEOUT VECTOR SERVICE
:ROUTINE ADDRESS FOR STACK VIO-
:LATION

```

010136 012737 000340 000006

MOV #340,2#ERRVEC+2

:NEW PS ON TIMEOUT TRAP

```

3070 010144 012737 010156 001332      MOV      #.+12,2#SGDDAT
3071
3072
3073 010152 005766 177742      TST      -36(SP)
3074
3075
3076
3077
3079 010156 104010      HLT      +10
3079
3080 010160 000421      BR       YELLOW
3081
3082 010162 104011      V1:     HLT      +11
3083
3084 010164 000417      BR       YELLOW
3085
3086 010166 042777 000001 171426 V2:     BIC      #BIT0,2#PARITY
3087 010174 005737 000372      TST      2#372
3088
3089 010200 001002      BNE     15
3090 010202 104012      HLT     +12
3091
3092 010204 000407      BR       YELLOW
3093
3094 010206 012706 001100      15:     MOV      #STACK,SP
3095 010212 013746 000372      MOV      2#372,-(SP)
3096
3097
3098 010216 004037 011550      JSR     RD,2#CHECKLOC
3099 010222 104003      HLT     +3
3100 010224 012706 001100      YELLOW: MOV      #STACK,SP
3101 010230 013737 010126 000004      MOV      2#SAVLOC4,2#4
3102 010236 005037 000006      CLR     2#6
3103 010242 005037 000372      CLR     2#372
3104
3105 010246 005037 000340      CLR     2#340
3106
3107
3108
3109
3110

```

```

:STORE THE PC THAT SHOULD
:BE PUSHED ON THE STACK
:IF A PARITY ABORT OCCURS
:REFERENCE BOTTOM LIMIT OF
:'YELLOW' ZONE USING REGISTER 6
:THIS INSTRUCTION SHOULD CAUSE
:AN ABORT 1ST, THEN A STACK
:VIOLATION
:DIDN'T ABORT OR RECOGNIZE THE
:STACK VIOLATION
:GO TO RESET STACK AND RESTORE
:TIMEOUT VECTORS
:ABORTED BUT STACK VIOLATION
:NOT RECOGNIZED
:GO TO RESET STACK AND RESTORE
:TIMEOUT VECTORS
:DISABLE PARITY
:STACK VIOLATION PICKED UP -
:WAS THE PARITY ABORT?
:BRANCH IF YES
:STACK VIOLATION PICKED UP
: BUT ABORT NOT RECOGNIZED
:GO TO RESET STACK AND RESTORE
:TIMEOUT VECTORS
:RESET STACK BACK TO NORMAL
:PUSH ABORT PC ONTO KERNAL
:STACK OUT OF VIOLATION AREAS
:FOR CHECKING PURPOSES
:CHECK FOR GOOD ABORT
:ABORTED INCORRECTLY
:RESET STACK BACK TO NORMAL
:RESTORE CONTENTS OF LOC. 4
:RESTORE CONTENTS OF LOC. 6
:RESET TRAPCATCHER LOCATION
:FOR NEXT TEST
:CLEAR BOTTOM LIMIT LOCATION
:OF 'YELLOW' ZONE USING NORMAL
:PARITY

```

```

*****
*****
*****
*****

```

```

:THE FOLLOWING TEST WILL/SHOULD CAUSE A PARITY ABORT IN THE
:'RED' ZONE. THE 'RED' ZONE IS THE AREA BEYOND THE 'YELLOW'
:ZONE DESCRIBED IN THE ABOVE TEST. I.E. LOCATIONS 336 ON
:DOWN COMPRISE THE 'RED' ZONE

```

```

:SINCE PARITY ERRORS HAVE HIGHEST PRIORITY WE WILL BE LOOKING
:FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
:TRAP TO LOCATION 4.

```

: THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
: SHOULD BE AS FOLLOWS:

: LOC. 0 - PC FROM STACK VIOLATION
: LOC. 2 - PS FROM STACK VIOLATION

: NOTE: THE PS AND PC FROM THE STACK VIOLATION ARE IN LOCS. 0 & 2
: BECAUSE THE STACK POINTER (R6) IS REPOSITIONED TO LOC. 4
: (BY HARDWARE)!!
:

:LOC. 374 - PS FROM PARITY ABORT
:LOC. 372 - PC FROM PARITY ABORT

:NOTE: THE ABOVE CONTENTS WILL EXIST IN THE 2 DIFFERENT STACK
AREAS IF BOTH THE PARITY ERROR AND STACK VIOLATION WERE
RECOGNIZED. THE TEST BELOW WILL DIFFERENTIATE BETWEEN
WHICH OCCURRED FIRST.

3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192

010252 000004
010254 004015
010256 010336
010260 005037 000336
010264 042777 000004 171330
010272 012706 000376
010276 013727 000004
010302 000000
010304 012737 010342 000004
010312 012737 000340 000006
010320 012737 010332 001332
010326 005766 177740
010332 104010
010334 000421
010336 104011
010340 000417
010342 042777 000001 171252
010350 005737 000372
010354 001002

:TEST 57 TEST ABORT IN 'RED' ZONE

TST57: SCOPE
11/45 **** ROM STATE 177 ****
:
11/40 **** ROM STATE 267 ****
JSR RO, (R5)
V3
CLR 2#336
BIC #BIT2, 2#PARITY
MOV #376, 5#SP
MOV 2#4, (PC)+
HOLDLOC4: .WORD 0
MOV #V4, 2#ERRVEC
MOV #340, 2#ERRVEC+2
MOV #.+12, 2#SGDDAT
TST -40(SP)
HLT +10
BR RED
V3: HLT +11
BR RED
V4: BIC #BIT0, 2#PARITY
TST 2#372
BNE 15

:SET UP A PARITY VECTOR SERVICE
:ROUTINE ADDRESS
:CLEAR 1ST LOCATION OF 'RED'
:ZONE USING 'OTHER' PARITY
:WRITE NORMAL
:SET STACK IN 'YELLOW' AREA
:SAVE CONTENTS OF LOC. 4
:IN NEXT LOCATION
:ORIGINAL CONTENTS OF LOC. 4
:GO HERE
:SET UP A TIMEOUT VECTOR SERVICE
:ROUTINE ADDRESS FOR STACK
:VIOLATION
:NEW PS ON TIMEOUT TRAP
:STORE THE PC THAT SHOULD
:BE PUSHED ON THE STACK
:IF A PARITY ABORT OCCURS
:REFERENCE 1ST LOCATION OF 'RED'
:ZONE USING REGISTER 6
:THIS INSTRUCTION SHOULD CAUSE
:AN ABORT 1ST, THEN A STACK
:VIOLATION
:DIDN'T ABORT OR RECOGNIZE THE
:STACK VIOLATION
:GO TO RESET STACK AND RESTORE
:TIMEOUT VECTORS
:ABORTED BUT STACK VIOLATION
:NOT RECOGNIZED
:GO TO RESET STACK AND RESTORE
:TIMEOUT VECTORS
:DISABLE PARITY
:STACK VIOLATION PICKED UP -
:WAS THE PARITY ABORT?
:BRANCH IF YES

```

3193 010356 104012          HLT      +12          ;STACK VIOLATION PICKED UP BUT
3194                                ;ABORT NOT RECOGNIZED
3195 010360 000407          BR       RED          ;GO TO RESET STACK AND RESTORE
3196                                ;TIMEOUT VECTORS
3197 010362 012706 001100    15:     MOV      #STACK,SP ;RESET STACK BACK TO NORMAL
3198 010366 013746 000372    MOV      @#372,-(SP) ;PUSH ABORT PC ONTC KERNAL
3199                                ;STACK OUT OF VIOLATION AREAS
3200                                ;FOR CHECKING PURPOSES
3201 010372 004037 011550          JSR      RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
3202 010376 104003          HLT      +3          ;ABORTED INCORRECTLY
3203 010400 012706 001100    RED:     MOV      #STACK,SP ;RESET STACK BACK TO NORMAL
3204 010404 013737 010302 000004    MOV      @#HOLDLOC4,@#4 ;RESTORE CONTENTS OF LOC. 4
3205 010412 005037 000006    CLR      @#6         ;RESTORE CONTENTS OF LOC. 6
3206 010416 005037 000372    CLR      @#372      ;RESET TRAPCATCHER LOCATION
3207                                ;FOR NEXT TEST
3208 010422 005037 000336    CLR      @#336      ;CLEAR 1ST LOCATION OF 'RED'
3209                                ;ZONE USING NORMAL PARITY
3210                                ;*****
3211                                ;TEST 60          TEST (INDEX WORD) SMO,DM7 MOV INSTRUCTION
3212                                ;*****
3213 010426 000004          †ST60: SCOPE
3214                                ;
3215                                ;
3216                                ;
3217 010430 005037 001624    11/45 **** ROM STATE 7 ****
3218                                ;
3219                                ;
3220                                ;
3221 010434 004015          CLR      @#PSPCORZONES ;CLEAR PS, PC OR ZONES
3222 010436 010514          ;ABORT FLAG
3223 010440 042777 000004 171154    JSR      RO,(R5)      ;SET UP PARITY VECTOR SERVICE
3224 010446 010100          W         ;ROUTINE ADDRESS
3225 010450 012720 010070    BIC      #BIT2,@PARITY ;WRITE NORMAL
3226                                ;SET UP FOR A DATO
3227 010454 052777 000004 171140    MOV      R1,RO       ;MOVE THE INSTRUCTION
3228 010462 012720 177776    MOV      #10070,(RO)+ ;'MOV RO,@-2(RO)' TO PARITY
3229                                ;MEMORY AREA
3230 010466 042777 000004 171126    BIS      #BIT2,@PARITY ;WRITE OTHER PARITY
3231 010474 010037 001332    MOV      #-2,(RO)+   ;WRITE THE INDEX WORD IN NEXT
3232                                ;PARITY MEMORY AREA LOCATION
3233 010500 012710 000203    BIC      #BIT2,@PARITY ;WRITE NORMAL
3234                                ;STORE THE PC THAT SHOULD
3235                                ;BE PUSHED ON THE STACK
3236 010504 004360 177774          MOV      RO,@#SGDDAT ;IF A PARITY ABORT OCCURS
3237 010510 104001          ;MOVE 'RTS R3' INTO NEXT
3238 010512 000410          ;PARITY MEMORY AREA LOCATION
3239 010514 042777 000001 171100 W:     MOV      #203,(RO)   ;IN CASE WE DON'T ABORT
3240 010522 004037 011550    JSR      R3,-4(RO)   ;GO TO PARITY MEMORY AREA
3241 010526 104003          HLT      +3          ;DIDN'T ABORT
3242 010530 012706 001100    BR       .+22        ;GO TO NEXT TEST
3243                                ;DISABLE PARITY
3244                                ;CHECK FOR GOOD ABORT
3245                                ;ABORTED INCORRECTLY
3246 010534 000004          JSR      RO,@#CHECKLOC ;RESET THE STACK
3247                                ;*****
3248                                ;TEST 61          TEST (INDEX WORD) SM6,DM0 CMP INSTRUCTION
3249                                ;*****
3250 †ST61: SCOPE
3251                                ;
3252                                ;
3253                                ;
3254                                ;
3255                                ;
3256                                ;
3257                                ;
3258                                ;
3259                                ;
3260                                ;
3261                                ;
3262                                ;
3263                                ;
3264                                ;
3265                                ;
3266                                ;
3267                                ;
3268                                ;
3269                                ;
3270                                ;
3271                                ;
3272                                ;
3273                                ;
3274                                ;
3275                                ;
3276                                ;
3277                                ;
3278                                ;
3279                                ;
3280                                ;
3281                                ;
3282                                ;
3283                                ;
3284                                ;
3285                                ;
3286                                ;
3287                                ;
3288                                ;
3289                                ;
3290                                ;
3291                                ;
3292                                ;
3293                                ;
3294                                ;
3295                                ;
3296                                ;
3297                                ;
3298                                ;
3299                                ;
3300                                ;
3301                                ;
3302                                ;
3303                                ;
3304                                ;
3305                                ;
3306                                ;
3307                                ;
3308                                ;
3309                                ;
3310                                ;
3311                                ;
3312                                ;
3313                                ;
3314                                ;
3315                                ;
3316                                ;
3317                                ;
3318                                ;
3319                                ;
3320                                ;
3321                                ;
3322                                ;
3323                                ;
3324                                ;
3325                                ;
3326                                ;
3327                                ;
3328                                ;
3329                                ;
3330                                ;
3331                                ;
3332                                ;
3333                                ;
3334                                ;
3335                                ;
3336                                ;
3337                                ;
3338                                ;
3339                                ;
3340                                ;
3341                                ;
3342                                ;
3343                                ;
3344                                ;
3345                                ;
3346                                ;
3347                                ;
3348                                ;
3349                                ;
3350                                ;
3351                                ;
3352                                ;
3353                                ;
3354                                ;
3355                                ;
3356                                ;
3357                                ;
3358                                ;
3359                                ;
3360                                ;
3361                                ;
3362                                ;
3363                                ;
3364                                ;
3365                                ;
3366                                ;
3367                                ;
3368                                ;
3369                                ;
3370                                ;
3371                                ;
3372                                ;
3373                                ;
3374                                ;
3375                                ;
3376                                ;
3377                                ;
3378                                ;
3379                                ;
3380                                ;
3381                                ;
3382                                ;
3383                                ;
3384                                ;
3385                                ;
3386                                ;
3387                                ;
3388                                ;
3389                                ;
3390                                ;
3391                                ;
3392                                ;
3393                                ;
3394                                ;
3395                                ;
3396                                ;
3397                                ;
3398                                ;
3399                                ;
3400                                ;
3401                                ;
3402                                ;
3403                                ;
3404                                ;
3405                                ;
3406                                ;
3407                                ;
3408                                ;
3409                                ;
3410                                ;
3411                                ;
3412                                ;
3413                                ;
3414                                ;
3415                                ;
3416                                ;
3417                                ;
3418                                ;
3419                                ;
3420                                ;
3421                                ;
3422                                ;
3423                                ;
3424                                ;
3425                                ;
3426                                ;
3427                                ;
3428                                ;
3429                                ;
3430                                ;
3431                                ;
3432                                ;
3433                                ;
3434                                ;
3435                                ;
3436                                ;
3437                                ;
3438                                ;
3439                                ;
3440                                ;
3441                                ;
3442                                ;
3443                                ;
3444                                ;
3445                                ;
3446                                ;
3447                                ;
3448                                ;
3449                                ;
3450                                ;
3451                                ;
3452                                ;
3453                                ;
3454                                ;
3455                                ;
3456                                ;
3457                                ;
3458                                ;
3459                                ;
3460                                ;
3461                                ;
3462                                ;
3463                                ;
3464                                ;
3465                                ;
3466                                ;
3467                                ;
3468                                ;
3469                                ;
3470                                ;
3471                                ;
3472                                ;
3473                                ;
3474                                ;
3475                                ;
3476                                ;
3477                                ;
3478                                ;
3479                                ;
3480                                ;
3481                                ;
3482                                ;
3483                                ;
3484                                ;
3485                                ;
3486                                ;
3487                                ;
3488                                ;
3489                                ;
3490                                ;
3491                                ;
3492                                ;
3493                                ;
3494                                ;
3495                                ;
3496                                ;
3497                                ;
3498                                ;
3499                                ;
3500                                ;
3501                                ;
3502                                ;
3503                                ;
3504                                ;
3505                                ;
3506                                ;
3507                                ;
3508                                ;
3509                                ;
3510                                ;
3511                                ;
3512                                ;
3513                                ;
3514                                ;
3515                                ;
3516                                ;
3517                                ;
3518                                ;
3519                                ;
3520                                ;
3521                                ;
3522                                ;
3523                                ;
3524                                ;
3525                                ;
3526                                ;
3527                                ;
3528                                ;
3529                                ;
3530                                ;
3531                                ;
3532                                ;
3533                                ;
3534                                ;
3535                                ;
3536                                ;
3537                                ;
3538                                ;
3539                                ;
3540                                ;
3541                                ;
3542                                ;
3543                                ;
3544                                ;
3545                                ;
3546                                ;
3547                                ;
3548                                ;
3549                                ;
3550                                ;
3551                                ;
3552                                ;
3553                                ;
3554                                ;
3555                                ;
3556                                ;
3557                                ;
3558                                ;
3559                                ;
3560                                ;
3561                                ;
3562                                ;
3563                                ;
3564                                ;
3565                                ;
3566                                ;
3567                                ;
3568                                ;
3569                                ;
3570                                ;
3571                                ;
3572                                ;
3573                                ;
3574                                ;
3575                                ;
3576                                ;
3577                                ;
3578                                ;
3579                                ;
3580                                ;
3581                                ;
3582                                ;
3583                                ;
3584                                ;
3585                                ;
3586                                ;
3587                                ;
3588                                ;
3589                                ;
3590                                ;
3591                                ;
3592                                ;
3593                                ;
3594                                ;
3595                                ;
3596                                ;
3597                                ;
3598                                ;
3599                                ;
3600                                ;
3601                                ;
3602                                ;
3603                                ;
3604                                ;
3605                                ;
3606                                ;
3607                                ;
3608                                ;
3609                                ;
3610                                ;
3611                                ;
3612                                ;
3613                                ;
3614                                ;
3615                                ;
3616                                ;
3617                                ;
3618                                ;
3619                                ;
3620                                ;
3621                                ;
3622                                ;
3623                                ;
3624                                ;
3625                                ;
3626                                ;
3627                                ;
3628                                ;
3629                                ;
3630                                ;
3631                                ;
3632                                ;
3633                                ;
3634                                ;
3635                                ;
3636                                ;
3637                                ;
3638                                ;
3639                                ;
3640                                ;
3641                                ;
3642                                ;
3643                                ;
3644                                ;
3645                                ;
3646                                ;
3647                                ;
3648                                ;
3649                                ;
3650                                ;
3651                                ;
3652                                ;
3653                                ;
3654                                ;
3655                                ;
3656                                ;
3657                                ;
3658                                ;
3659                                ;
3660                                ;
3661                                ;
3662                                ;
3663                                ;
3664                                ;
3665                                ;
3666                                ;
3667                                ;
3668                                ;
3669                                ;
3670                                ;
3671                                ;
3672                                ;
3673                                ;
3674                                ;
3675                                ;
3676                                ;
3677                                ;
3678                                ;
3679                                ;
3680                                ;
3681                                ;
3682                                ;
3683                                ;
3684                                ;
3685                                ;
3686                                ;
3687                                ;
3688                                ;
3689                                ;
3690                                ;
3691                                ;
3692                                ;
3693                                ;
3694                                ;
3695                                ;
3696                                ;
3697                                ;
3698                                ;
3699                                ;
3700                                ;
3701                                ;
3702                                ;
3703                                ;
3704                                ;
3705                                ;
3706                                ;
3707                                ;
3708                                ;
3709                                ;
3710                                ;
3711                                ;
3712                                ;
3713                                ;
3714                                ;
3715                                ;
3716                                ;
3717                                ;
3718                                ;
3719                                ;
3720                                ;
3721                                ;
3722                                ;
3723                                ;
3724                                ;
3725                                ;
3726                                ;
3727                                ;
3728                                ;
3729                                ;
3730                                ;
3731                                ;
3732                                ;
3733                                ;
3734                                ;
3735                                ;
3736                                ;
3737                                ;
3738                                ;
3739                                ;
3740                                ;
3741                                ;
3742                                ;
3743                                ;
3744                                ;
3745                                ;
3746                                ;
3747                                ;
3748                                ;
3749                                ;
3750                                ;
3751                                ;
3752                                ;
3753                                ;
3754                                ;
3755                                ;
3756                                ;
3757                                ;
3758                                ;
3759                                ;
3760                                ;
3761                                ;
3762                                ;
3763                                ;
3764                                ;
3765                                ;
3766                                ;
3767                                ;
3768                                ;
3769                                ;
3770                                ;
3771                                ;
3772                                ;
3773                                ;
3774                                ;
3775                                ;
3776                                ;
3777                                ;
3778                                ;
3779                                ;
3780                                ;
3781                                ;
3782                                ;
3783                                ;
3784                                ;
3785                                ;
3786                                ;
3787                                ;
3788                                ;
3789                                ;
3790                                ;
3791                                ;
3792                                ;
3793                                ;
3794                                ;
3795                                ;
3796                                ;
3797                                ;
3798                                ;
3799                                ;
3800                                ;
3801                                ;
3802                                ;
3803                                ;
3804                                ;
3805                                ;
3806                                ;
3807                                ;
3808                                ;
3809                                ;
3810                                ;
3811                                ;
3812                                ;
3813                                ;
3814                                ;
3815                                ;
3816                                ;
3817                                ;
3818                                ;
3819                                ;
3820                                ;
3821                                ;
3822                                ;
3823                                ;
3824                                ;
3825                                ;
3826                                ;
3827                                ;
3828                                ;
3829                                ;
3830                                ;
3831                                ;
3832                                ;
3833                                ;
3834                                ;
3835                                ;
3836                                ;
3837                                ;
3838                                ;
3839                                ;
3840                                ;
3841                                ;
3842                                ;
3843                                ;
3844                                ;
3845                                ;
3846                                ;
3847                                ;
3848                                ;
3849                                ;
3850                                ;
3851                                ;
3852                                ;
3853                                ;
3854                                ;
3855                                ;
3856                                ;
3857                                ;
3858                                ;
3859                                ;
3860                                ;
3861                                ;
3862                                ;
3863                                ;
3864                                ;
3865                                ;
3866                                ;
3867                                ;
3868                                ;
3869                                ;
3870                                ;
3871                                ;
3872                                ;
3873                                ;
3874                                ;
3875                                ;
3876                                ;
3877                                ;
3878                                ;
3879                                ;
3880                                ;
3881                                ;
3882                                ;
3883                                ;
3884                                ;
3885                                ;
3886                                ;
3887                                ;
3888                                ;
3889                                ;
3890                                ;
3891                                ;
3892                                ;
3893                                ;
3894                                ;
3895                                ;
3896                                ;
3897                                ;
3898                                ;
3899                                ;
3900                                ;
3901                                ;
3902                                ;
3903                                ;
3904                                ;
3905                                ;
3906                                ;
3907                                ;
3908                                ;
3909                                ;
3910                                ;
3911                                ;
3912                                ;
3913                                ;
3914                                ;
3915                                ;
3916                                ;
3917                                ;
3918                                ;
3919                                ;
3920                                ;
3921                                ;
3922                                ;
3923                                ;
3924                                ;
3925                                ;
3926                                ;
3927                                ;
3928                                ;
3929                                ;
3930                                ;
3931                                ;
3932                                ;
3933                                ;
3934                                ;
3935                                ;
3936                                ;
3937                                ;
3938                                ;
3939                                ;
3940                                ;
3941                                ;
3942                                ;
3943                                ;
3944                                ;
3945                                ;
3946                                ;
3947                                ;
3948                                ;
3949                                ;
3950                                ;
3951                                ;
3952                                ;
3953                                ;
3954                                ;
3955                                ;
3956                                ;
3957                                ;
3958                                ;
3959                                ;
3960                                ;
3961                                ;
3962                                ;
3963                                ;
3964                                ;
3965                                ;
3966                                ;
3967                                ;
3968                                ;
3969                                ;
3970                                ;
3971                                ;
3972                                ;
3973                                ;
3974                                ;
3975                                ;
3976                                ;
3977                                ;
3978                                ;
3979                                ;
3980                                ;
3981                                ;
3982                                ;
3983                                ;
3984                                ;
3985                                ;
3986                                ;
3987                                ;
3988                                ;
3989                                ;
3990                                ;
3991                                ;
3992                                ;
3993                                ;
3994                                ;
3995                                ;
3996                                ;
3997                                ;
3998                                ;
3999                                ;
4000                                ;

```

```

3249      :      11/40 **** ROM STATE 241 ****
3250 010536 004015      JSR      RO, (R5)      ; SET UP PARITY VECTOR SERVICE
3251 010540 010616      XX      ; ROUTINE ADDRESS
3252 010542 042777 000004 171052      BIC      #BIT2, @PARITY      ; WRITE NORMAL
3253 010550 010100      MOV      R1, RO      ; SET UP FOR A DATO
3254 010552 012720 026000      MOV      #26000, (RO)+      ; MOVE THE INSTRUCTION
3255      ; 'CMP -2(RO), RO' TO PARITY
3256      ; MEMORY AREA
3257 010556 052777 000004 171036      BIS      #BIT2, @PARITY      ; WRITE OTHER PARITY
3258 010564 012720 177776      MOV      #-2, (RO)+      ; WRITE THE INDEX WORD IN NEXT
3259      ; PARITY MEMORY AREA LOCATION
3260 010570 042777 000004 171024      BIC      #BIT2, @PARITY      ; WRITE NORMAL
3261 010576 010037 001332      MOV      RO, @#$GDDAT      ; STORE THE PC THAT SHOULD
3262      ; BE PUSHED ON THE STACK
3263      ; IF A PARITY ABORT OCCURS
3264 010602 012710 000203      MOV      #203, (RO)      ; MOVE 'RTS R3' INTO NEXT
3265      ; PARITY MEMORY AREA LOCATION
3266      ; IN CASE WE DON'T ABORT
3267 010606 004360 177774      JSR      R3, -4(RO)      ; GO TO PARITY MEMORY AREA
3268 010612 104001      HLT      +1      ; DIDN'T ABORT
3269 010614 000410      BR      .+22      ; GO TO NEXT TEST
3270 010616 042777 000001 170776 XX:      BIC      #BIT0, @PARITY      ; DISABLE PARITY
3271 010624 004037 011550      JSR      RO, @#$CHECKLOC      ; CHECK FOR GOOD ABORT
3272 010630 104003      HLT      +3      ; ABORTED INCORRECTLY
3273 010632 012706 001100      MOV      #STACK, SP      ; RESET THE STACK
3274      ; *****
3275      ; TEST 62      TEST (INDEX WORD) SMO, DM6 MOV INSTRUCTION
3276      ; *****
3277 010636 000004      ST62: SCOPE
3278      ;
3279      ;
3280      ;      11/45 **** ROM STATE 6 ****
3281      ;
3282      ;      11/40 **** ROM STATE 206 ****
3281 010640 004015      JSR      RO, (R5)      ; SET UP PARITY VECTOR SERVICE
3282 010642 010720      Y      ; ROUTINE ADDRESS
3283 010644 042777 000004 170750      BIC      #BIT2, @PARITY      ; WRITE NORMAL
3284 010652 010100      MOV      R1, RO      ; SET UP FOR A DATO
3285 010654 012720 010060      MOV      #10060, (RO)+      ; MOVE THE INSTRUCTION
3286      ; 'MOV RO, -2(RO)' TO PARITY
3287      ; MEMORY AREA
3288 010660 052777 000004 170734      BIS      #BIT2, @PARITY      ; WRITE OTHER PARITY
3289 010666 012720 177776      MOV      #-2, (RO)+      ; WRITE THE INDEX WORD IN NEXT
3290      ; PARITY MEMORY AREA LOCATION
3291 010672 042777 000004 170722      BIC      #BIT2, @PARITY      ; WRITE NORMAL
3292 010700 010037 001332      MOV      RO, @#$GDDAT      ; STORE THE PC THAT SHOULD
3293      ; BE PUSHED ON THE STACK
3294      ; IF A PARITY ABORT OCCURS
3295 010704 012710 000203      MOV      #203, (RO)      ; MOVE 'RTS R3' INTO NEXT
3296      ; PARITY MEMORY AREA LOCATION
3297      ; IN CASE WE DON'T ABORT
3298 010710 004360 177774      JSR      R3, -4(RO)      ; GO TO PARITY MEMORY AREA
3299 010714 104001      HLT      +1      ; DIDN'T ABORT
3300 010716 000410      BR      .+22      ; GO TO NEXT TEST
3301 010720 042777 000001 170674 Y:      BIC      #BIT0, @PARITY      ; DISABLE PARITY
3302 010726 004037 011550      JSR      RO, @#$CHECKLOC      ; CHECK FOR GOOD ABORT
3303 010732 104003      HLT      +3      ; ABORTED INCORRECTLY
3304 010734 012706 001100      MOV      #STACK, SP      ; RESET THE STACK

```

```

3305
3306
3307
3308 010740 000004
3309
3310
3311
3312 010742 004015
3313 010744 011056
3314 010746 042777 000004 170646
3315 010754 010102
3316 010756 162702 000004
3317
3318 010762 012722 012700
3319
3320
3321 010766 012722 177777
3322
3323 010772 012722 100001
3324
3325
3326 010776 052777 000004 170616
3327 011004 005737 001642
3328 011010 001405
3329
3330
3331 011012 010237 001332
3332
3333
3334
3335
3336 011016 012722 000240
3337
3338 011022 000404
3339 011024 012722 000240 3$:
3340
3341 011030 010237 001332
3342
3343
3344 011034 042777 000004 170560 1$:
3345 011042 012712 000203
3346
3347
3348 011046 004362 177770
3349 011052 104001
3350 011054 000410
3351 011056 042777 000001 170536 20:
3352 011064 004037 011550
3353 011070 104003
3354 011072 012706 001100
3355
3356
3357 011076 000004
3358
3359
3360

```

```

:*****
:TEST 63 TEST BPL INSTRUCTION (-BCOK)
:*****
↑ST63: SCOPE
:
: 11/45 **** ROM STATE 321 ****
:
: 11/40 **** ROM STATE 1 ****
: JSR RO, (R5)
: Z0
: BIC #BIT2, @PARITY
: MOV R1, R2
: SUB #4, R2
:
: MOV #12700, (R2)+
:
: MOV #-1, (R2)+
:
: MOV #100001, (R2)+
:
: BIS #BIT2, @PARITY
: TST @#CPU40
: BEQ 2$
:
: MOV R2, @#$GDDAT
:
:
: MOV #240, (R2)+
: BR 1$
: MOV #240, (R2)+
:
: MOV R2, @#$GDDAT
:
: BIC #BIT2, @PARITY
: MOV #203, (R2)
:
: JSR R3, -10(R2)
: HLT +1
: BR .+22
: BIC #BIT0, @PARITY
: JSR RO, @#CHECKLOC
: HLT +3
: MOV #STACK, SP
:*****
:TEST 64 TEST BNE INSTRUCTION (-BCOK)
:*****
↑ST64: SCOPE
:
: 11/45 **** ROM STATE 322 ****
:

```

```

: SET UP PARITY VECTOR SERVICE
: ROUTINE ADDRESS
: WRITE NORMAL
: SET UP FOR A DATO
: CALCULATE THE START
: ADDRESS FOR THIS TEST
: MOVE THE INSTRUCTION
: 'MOV #-1, RO' TO PARITY
: MEMORY AREA
: MOVE A -1 INTO NEXT
: PARITY MEMORY AREA LOCATION
: MOVE THE INSTRUCTION
: 'BPL .+4' INTO NEXT PARITY
: MEMORY AREA LOCATION
: WRITE OTHER PARITY
: ARE WE ON AN 11/40?
: BRANCH IF NO
: SINCE THE PC WILL BE UPDATED
: ON THE PARITY ABORT
: STORE THIS PC THAT SHOULD
: BE PUSHED ON THE STACK
: IF A PARITY ABORT OCCURS
: SINCE THE PC IS NOT UPDATED
: ON THE PARITY ABORT
: MOVE A 'NOP' INTO NEXT
: PARITY MEMORY AREA LOCATION
: CONTINUE WITH TEST
: MOVE A 'NOP' INTO NEXT
: PARITY MEMORY AREA LOCATION
: STORE THE PC THAT SHOULD
: BE PUSHED ON THE STACK
: IF A PARITY ABORT OCCURS
: WRITE NORMAL
: MOVE 'RTS R3' INTO NEXT
: PARITY MEMORY AREA LOCATION
: IN CASE WE DON'T ABORT
: GO TO PARITY MEMORY AREA
: DIDN'T ABORT
: GO TO NEXT TEST
: DISABLE PARITY
: CHECK FOR GOOD ABORT
: ABORTED INCORRECTLY
: RESET THE STACK

```

```

3361 ; 11/40 **** ROM STATE 1 ****
3362 011100 004015 JSR RO,(R5) ;SET UP PARITY VECTOR SERVICE
3363 011102 011214 Z1 ;ROUTINE ADDRESS
3364 011104 042777 000004 170510 BIC #BIT2,@PARITY ;WRITE NORMAL
3365 011112 010102 MOV R1,R2 ;SET UP FOR A DATO
3366 011114 162702 000004 SUB #4,R2 ;CALCULATE THE START ADDRESS
3367 ;FOR THIS TEST
3368 011120 012722 012700 MOV #12700,(R2)+ ;MOVE THE INSTRUCTION
3369 ;'MOV #0,R0' INTO PARITY
3370 ;MEMORY AREA
3371 011124 012722 000000 MOV #0,(R2)+ ;MOVE A 0 INTO NEXT PARITY
3372 ;MEMORY AREA LOCATION
3373 011130 012722 001001 MOV #1001,(R2)+ ;MOVE THE INSTRUCTION
3374 ;'BNE .+4' INTO NEXT
3375 ;PARITY MEMORY AREA LOCATION
3376 011134 052777 000004 170460 BIS #BIT2,@PARITY ;WRITE OTHER PARITY
3377 011142 005737 001642 TST @#CPU40 ;ARE WE ON AN 11/40?
3378 011146 001405 BEQ 2$ ;BRANCH IF NO
3379 ;SINCE THE PC WILL BE UPDATED
3380 ;ON THE PARITY ABORT
3381 011150 010237 001332 MOV R2,@#$GDDAT ;STORE THIS PC THAT SHOULD
3382 ;BE PUSHED ON THE STACK
3383 ;IF A PARITY ABORT OCCURS
3384 ;SINCE THE PC WILL NOT BE
3385 ;UPDATED ON THE PARITY ABORT
3386 011154 012722 000240 MOV #240,(R2)+ ;MOVE A 'NOP' INTO NEXT
3387 ;PARITY MEMORY AREA LOCATION
3388 011160 000404 BR 1$ ;CONTINUE WITH TEST
3389 011162 012722 000240 2$: MOV #240,(R2)+ ;MOVE A 'NOP' INTO NEXT
3390 ;PARITY MEMORY AREA LOCATION
3391 011166 010237 001332 MOV R2,@#$GDDAT ;STORE THE PC THAT SHOULD
3392 ;BE PUSHED ON THE STACK
3393 ;IF A PARITY ABORT OCCURS
3394 011172 042777 000004 170422 1$: BIC #BIT2,@PARITY ;WRITE NORMAL
3395 011200 012712 000203 MOV #203,(R2) ;MOVE 'RTS R3' INTO NEXT
3396 ;PARITY MEMORY AREA LOCATION
3397 ;IN CASE WE DON'T ABORT
3398 011204 004362 177770 JSR R3,-10(R2) ;GO TO PARITY MEMORY AREA
3399 011210 104001 HLT +1 ;DIDN'T ABORT
3400 011212 000410 BR .+22 ;GO TO NEXT TEST
3401 011214 042777 000001 170400 Z1: BIC #BIT0,@PARITY ;DISABLE PARITY
3402 011222 004037 011550 JSR RO,@#CHECKLOC ;CHECK FOR GOOD ABORT
3403 011226 104003 HLT +3 ;ABORTED INCORRECTLY
3404 011230 012706 001100 MOV #STACK,SP ;RESET THE STACK
3405 ;*****
3406 ;TEST 65 TEST BEQ INSTRUCTION (-BCOK)
3407 ;*****
3408 011234 000004 †ST65: SCOPE
3409 ;
3410 ;
3411 ; 11/45 **** ROM STATE 324 ****
3412 ;
3413 ; 11/40 **** ROM STATE 1 ****
3414 011236 004015 JSR RO,(R5) ;SET UP PARITY VECTOR SERVICE
3415 011240 011352 Z2 ;ROUTINE ADDRESS
3416 011242 042777 000004 170352 BIC #BIT2,@PARITY ;WRITE NORMAL
3417 011250 010102 MOV R1,R2 ;SET UP FOR A DATO
3418 011252 162702 000004 SUB #4,R2 ;CALCULATE THE START ADDRESS

```



```

3417
3418 011256 012722 012700      MOV      #12700,(R2)+
3419
3420
3421 011262 012722 177777      MOV      #-1,(R2)+
3422
3423 011266 012722 001401      MOV      #1401,(R2)+
3424
3425
3426 011272 052777 000004 170322  BIS      #BIT2,@PARITY
3427 011300 005737 001642      TST      @#CPU40
3428 011304 001405      BEQ      2$
3429
3430
3431 011306 010237 001332      MOV      R2,@#$GDDAT
3432
3433
3434
3435
3436 011312 012722 000240      MOV      #240,(R2)+
3437
3438 011316 000404      BR      1$
3439 011320 012722 000240      2$: MOV      #240,(R2)+
3440
3441 011324 010237 001332      MOV      R2,@#$GDDAT
3442
3443
3444 011330 042777 000004 170264  1$: BIC      #BIT2,@PARITY
3445 011336 012712 000203      MOV      #203,(R2)
3446
3447
3448 011342 004362 177770      JSR      R3,-10(R2)
3449 011346 104001      HLT      +1
3450 011350 000410      BR      .+22
3451 011352 042777 000001 170242  22: BIC      #BIT0,@PARITY
3452 011360 004037 011550      JSR      R0,@#CHECKLOC
3453 011364 104003      HLT      +3
3454 011366 012706 001100      MOV      #STACK,SP
3455
3456
3457
3458 011372 000004      ;*****
3459 011374 042777 100005 170220  ;TEST 66      END OF PROGRAM
3460
3461 011402 005737 001630      ;*****
3462 011406 001014      ;TST66: SCOPE
3463
3464 011410 062737 000002 001336  BIC      #BIT15!BIT2!BIT0,@PARITY      ;DISABLE ALL PARITY
3465 011416 062737 000002 001414      ;AND CLEAR ERROR BIT!
3466
3467 011424 062737 000002 001416  TST      @#USERTYPE      ;DID THE USER SELECT THE REGISTER?
3468
3469 011432 062737 000002 001446  BNE      1$      ;BRANCH IF SO AND DON'T STEP
3470
3471 011440 004337 011474      ;UP THE TABLE
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499

```



```

3595                                     : (BITS 5 THRU 11)
3596 011636 042704 170037                BIC    #170037,R4      : CLEAR ALL BITS EXCEPT 5 THRU 11
3597 011642 020402                      CMP    R4,R2          : ARE THE ERROR ADDRESS BITS
3598                                     : WHAT THEY SHOULD BE?
3599 011644 001026                        BNE    4$            : BRANCH IF NO
3600 011646 010237 001326                7$:  MOV    R2,2#$GDADR  : STORE WHAT THE ADDRESS BITS
3601                                     : SHOULD HAVE BEEN
3602 011652 010437 001330                MOV    R4,2#$BDADR  : STORE WHAT THE ADDRESS BITS
3603                                     : WERE
3604 011656 012604                      MOV    (SP)+,R4     : RESTORE R4 CONTENTS
3605 011660 012603                      MOV    (SP)+,R3     : RESTORE R3 CONTENTS
3606 011662 012602                      MOV    (SP)+,R2     : RESTORE R2 CONTENTS
3607                                     : IF WE HAVE REACHED THIS PATH 1 OF 2 CONDITIONS EXIST --
3608                                     : WE HAVE AN OLD MOS DESIGN WITH NO ADDRESS BITS, OR WE HAVE
3609                                     : CORE OR THE NEW MOS DESIGN WITH ADDRESS BITS OK!!
3610 011664 026637 000002 001332        CMP    2(SP),2#$GDAT : WAS THE CORRECT PC PUSHED
3611                                     : ON THE STACK?
3612 011672 001404                      BEQ    9$            : BRANCH IF YES
3613 011674 016637 000002 001334        MOV    2(SP),2#$BDAT : SAVE INCORRECT PC FOR PRINTOUT
3614 011702 000200                      RTS    R0            : GO BACK TO INDICATE BAD ABORT
3615 011704 005720                      9$:  TST    (R0)+      : STEP UP RETURN ADDRESS
3616                                     : TO BYPASS THE ERROR HLT
3617 011706 000200                      RTS    R0            : RETURN TO CONTINUE TESTING
3618 011710 062703 004000                3$:  ADD    #4000,R3   : STEP UP TO THE NEXT POSSIBLE
3619                                     : ABORT LOCATION AREA
3620 011714 062702 000040                ADD    #40,R2        : CHANGE THE VALUE OF THE HIGH
3621                                     : ORDER ADDRESS BITS VALUE TO BE
3622                                     : CHECKED
3623 011720 000742                      BR     1$            : GO BACK TO CHECK THIS ONE
3624 011722 010237 001326                4$:  MOV    R2,2#$GDADR  : STORE WHAT THE ADDRESS BITS
3625                                     : SHOULD HAVE BEEN
3626 011726 010437 001330                MOV    R4,2#$BDADR  : STORE WHAT THE
3627                                     : ADDRESS BITS WERE
3628 011732 012604                      MOV    (SP)+,R4     : RESTORE R4 CONTENTS
3629 011734 012603                      MOV    (SP)+,R3     : RESTORE R3 CONTENTS
3630 011736 012602                      MOV    (SP)+,R2     : RESTORE R2 CONTENTS
3631                                     : IF WE HAVE REACHED THIS PATH WE HAVE CORE OR THE NEW MOS DESIGN
3632                                     : WITH BAD ADDRESS BITS WHICH SHOULD INDICATE AN IMPROPER ABORT!!
3633 011740 026637 000002 001332        CMP    2(SP),2#$GDAT : WAS THE CORRECT PC PUSHED
3634                                     : ON THE STACK?
3635 011746 001404                      BEQ    10$           : BRANCH IF YES
3636 011750 016637 000002 001334        MOV    2(SP),2#$BDAT : SAVE INCORRECT PC FOR PRINTOUT
3637 011756 000200                      RTS    R0            : GO BACK TO INDICATE BAD ABORT
3638 011760 016737 167346 001334        10$: MOV    $GDAT,2#$BDAT : RECORD THE FACT THAT THE
3639                                     : CORRECT PC WAS PUSHED ON STACK
3640 011766 000200                      RTS    R0            : GO BACK TO INDICATE BAD ABORT
3641
3642
3643
3644
3645

```

```

3636
3637
3638
3639
3640
3641
3642 011770 000004
3643 011772 005067 167304
3644 011776 005067 000306
3645 012002 005267 157272
3646 012006 032737 002000 177570
3647 012014 001002
3648 012016 104400 012056
3649 012022 013700 000042
3650 012026 001411
3651 012030 022760 177777 000002
3652 012036 001001
3653 012040 000005
3654 012042 004710
3655 012044 000240
3656 012046 000240
3657 012050 000240
3658 012052 000137 003620
3659 012056 177507 000377

```

```

:*****
:END OF PASS ROUTINE
:INCREMENT THE PASS NUMBER
:IF SW10=0 DING THE TTY BELL ON END OF PROGRAM
:IF THERE IS A MONITOR GO TO IT.
:IF NONE JUMP TO START
$EOP: SCOPE
      CLR $STNM ;ZERO THE TEST NUMBER
      CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
      INC $PASS ;INCREMENT THE PASS NUMBER
      BIT $SW10,$SWR ;RING THE BELL?
      BNE 4$ ;NO
      TYPE $BELL ;RING A DING
      MOV $42,R0 ;GET MONITOR ADDRESS
      BEQ $DOAGN ;IF NONE
      CMP #-1,2(R0)
      BNE $SENDAD
$SENDAD: JSR PC,(R0) ;GO TO MONITOR
         NOP ;SAVE ROOM
         NOP ;FOR
         NOP ;ACT11
$DOAGN: JMP $START ;RETURN
$BELL: .ASCIZ <207><377><377>

```

```

3660
3661
3662
3663
3664
3665
3666
3667
3668 012062
3669 012062 006137 177570
3670 012066 100502
3671
3672 012070 000416
3673
3674 012072 013746 000004
3675 012076 012737 012116 000004
3676 012104 005737 177060
3677 012110 012637 000004
3678 012114 000457
3679 012116 022626
3680 012120 012637 000004
3681 012124 000417
3682 012126
3683 012126 032737 000400 177570
3684 012134 001404
3685 012136 123767 177570 167136
3686 012144 001453
3687 012146 105767 167142
3688 012152 001415
3689 012154 032737 001000 177570
3690 012162 001404
3691 012164 016767 167120 167114
3692 012172 000440
3693 012174 105067 167114
3694 012200 005067 000104
3695 012204 000415
3696 012206 032737 004000 177570
3697 012214 001011
3698 012216 005767 167056
3699 012222 001406
3700 012224 005267 167054
3701 012230 026767 000054 167046
3702 012236 002016
3703 012240 012767 000001 167036
3704 012246 016767 000040 000034
3705 012254 005267 167022
3706 012260 011667 167022
3707 012264 011667 167020
3708 012270 005067 000500
3709 012274 016737 167002 177570
3710 012302 016716 167000
3711 012306 000002
3712 012310 000000
3713 012312 000000

```

```

;*****
;THIS ROUTINE IS THE SCOPE HANDLER
;SW14=1 LOOP ON TEST
;SW11=1 INHIBIT ITERATIONS
;SW09=1 LOOP ON ERROR
;SW08=1 LOOP ON TEST IN SW<7:0>
;THE TEST NUMBER ($STNM) IS UPDATED AND DISPLAYED
$SCOPE:
      ROL      2#SWR      ;LOOP ON PRESENT TEST?
      BMI      $OVER     ;YES IF SW14=1
      ;*****START OF CODE FOR THE XOR TESTER*****
      $XTSTR: BR      6$
      ;IF RUNNING ON THE "XOR" TESTER CHANGE
      ;THIS INSTRUCTION TO A "NOP" (NOP=240)
      ;SAVE THE CONTENTS OF THE ERROR VECTOR
      MOV      2#ERRVEC, -(SP)
      MOV      2$,$2#ERRVEC
      TST      2#177060
      ;SET FOR TIMEOUT
      ;TIME OUT ON XOR?
      MOV      (SP)+, 2#ERRVEC
      ;RESTORE THE ERROR VECTOR
      BR      $$VLAD
      ;GO TO THE NEXT TEST
      5$: CMP      (SP)+, (SP)+
      ;CLEAR THE STACK AFTER A TIME OUT
      MOV      (SP)+, 2#ERRVEC
      ;RESTORE THE ERROR VECTOR
      BR      7$
      ;LOOP ON THE PRESENT TEST
      6$: ;*****END OF CODE FOR THE XOR TESTER*****
      BIT      2#SW08, 2#SWR
      ;LOOP ON SPEC. TEST?
      BEQ      2$
      ;BR IF NO
      CMPB     2#SWR, $STNM
      ;ON THE RIGHT TEST? SWR<7:0>
      BEQ      $OVER
      ;BR IF YES
      2$: TSTB     $ERFLG
      ;HAS AN ERROR OCCURRED?
      BEQ      3$
      ;BR IF NO
      BIT      2#SW09, 2#SWR
      ;LOOP ON ERROR?
      BEQ      4$
      ;BR IF NO
      7$: MOV      $LPERR, $LPADR
      ;SET LOOP ADDRESS TO LAST SCOPE
      BR      $OVER
      4$: CLRB     $ERFLG
      ;ZERO THE ERROR FLAG
      CLR      $TIMES
      ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
      BR      1$
      ;ESCAPE TO THE NEXT TEST
      3$: BIT      2#SW11, 2#SWR
      ;INHIBIT ITERATIONS?
      BNE      1$
      ;BR IF YES
      TST      $PASS
      ;IF FIRST PASS OF PROGRAM
      BEQ      1$
      ;INHIBIT ITERATIONS
      INC      $ICNT
      ;INCREMENT ITERATION COUNT
      CMP      $TIMES, $ICNT
      ;CHECK THE NUMBER OF ITERATIONS MADE
      BGE      $OVER
      ;BR IF MORE ITERATION REQUIRED
      1$: MOV      #1, $ICNT
      ;REINITIALIZE THE ITERATION COUNTER
      MOV      $MXCNT, $TIMES
      ;SET NUMBER OF ITERATIONS TO DO
      $SVLAD: INC      $STNM
      ;COUNT TEST NUMBERS
      MOV      (SP), $LPADR
      ;SAVE SCOPE LOOP ADDRESS
      MOV      (SP), $LPERR
      ;SAVE ERROR LOOP ADDRESS
      CLR      $ESCAPE
      ;CLEAR THE ESCAPE FROM ERROR ADDRESS
      $OVER: MOV      $STNM, 2#DISPLAY
      ;DISPLAY TEST NUMBER
      MOV      $LPADR, (SP)
      ;FUDGE RETURN ADDRESS
      RTI
      ;FIXES PS
      $TIMES: 0
      ;NUMBER OF ITERATIONS TO PERFORM
      $MXCNT: 0
      ;MAX. NUMBER OF ITERATIONS

```

```

3714
3715
3716
3717
3718
3719 012314
3720 012314 010046
3721 012316 010146
3722 012320 010246
3723 012322 010346
3724 012324 016600 000010
3725 012330 005001
3726 012332 012702 000006
3727 012336 104402
3728 012340 112603
3729 012342 110367 000102
3730 012346 104400 012450
3731 012352 022703 000015
3732 012356 001422
3733 012360 022703 000060
3734 012364 003014
3735 012366 022703 000067
3736 012372 002411
3737 012374 005302
3738 012376 002407
3739 012400 006301
3740 012402 006301
3741 012404 006301
3742 012406 042703 177770
3743 012412 050301
3744 012414 000750
3745 012416 104400 012452
3746 012422 000742
3747 012424 104400 012454
3748 012430 010130
3749 012432 010066 000010
3750 012436 012603
3751 012440 012602
3752 012442 012601
3753 012444 012600
3754 012446 000002
3755 012450 000 000
3756 012452 077
3757 012453 015
3758 012454 000012
3759
3760
3761
3762
3763
3764
3765
3766
3767 012456 011646
3768 012460 016666 000004 000002
3769 012466 105777 000126

```

```

:*****
:ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
:CALL:
:      ACCEPT ,ADDR          ;PUT OCTAL NUMBER IN ADDR
$ACCEPT:
      MOV      RO,-(SP)      ;PUSH RO ON STACK
      MOV      R1,-(SP)      ;PUSH R1 ON STACK
      MOV      R2,-(SP)      ;PUSH R2 ON STACK
      MOV      R3,-(SP)      ;PUSH R3 ON STACK
      MOV      10(SP),RO     ;GET ADDRESS POINTER OF WHERE TO PUT NUMBER
1$:   CLR      R1            ;CLEAR PARTIAL NUMBER
      MOV      #6,R2         ;MAX. # OF DIGITS ALLOWED
2$:   GETCHR          ;GET ONE CHARACTER
      MOV      (SP)+,R3      ;AND PUT IT IN R3
      MOV      R3,6$         ;ECHO THE CHARACTER
      TYPE     6$           ;WAS THIS CHARACTER A "CR"?
      BEQ      #15,R3        ;BR IF YES
      CMP      #'0,R3        ;INSURE THE CHARACTER IS
      BGT      4$           ;A DIGIT BETWEEN 0 AND 7.
      CMP      #'7,R3
      BLT      4$
      DEC      R2            ;CHECK NUMBER OF CHARACTERS
      BLT      4$           ;BR IF TO MANY
      ASL      R1            ;POSITION PARTIAL NUMBER
      ASL      R1            ;FOR THIS DIGIT
      BIC      #1C(7),R3     ;GET RID OF THE ASCII JUNK
      BIS      R3,R1         ;COMBINE THIS DIGIT WITH PARTIAL
      BR       2$           ;GO GET ANOTHER DIGIT
4$:   TYPE     $QUES         ;TYPE "?"
      BR       1$           ;GO START OVER
5$:   TYPE     $LF           ;FOLLOW "CR" WITH A "LF"
      MOV      R1,2(RO)+     ;PASS THE NUMBER TO THE USER
      MOV      RO,10(SP)     ;SET FOR RETURN
      MOV      (SP)+,R3      ;POP STACK INTO R3
      MOV      (SP)+,R2      ;POP STACK INTO R2
      MOV      (SP)+,R1      ;POP STACK INTO R1
      MOV      (SP)+,RO      ;POP STACK INTO RO
      RTI
6$:   .BYTE    0,0          ;STORAGE FOR ASCII CHAR. AND TERMINATOR
$QUES: .ASCII  '?'         ;QUESTION MARK
$CRLF: .ASCII  <15        ;CARRIAGE RETURN
$LF:   .ASCII  <12        ;LINEFEED
      ;SELECTED REGISTER FROM THE TTY
:*****
:THIS ROUTINE INPUTS A SINGLE CHARACTER FROM THE TTY
:CALL:
:      GETCHR          ;INPUT A SINGLE CHARACTER FROM THE TTY
:      RETURN HERE    ;CHARACTER IS ON THE STACK
$READC: MOV      (SP),-(SP)  ;PUSH DOWN THE PC
      MOV      4(SP),2(SP)  ;SAVE THE PS
1$:   TSTB     @TKS         ;WAIT FOR

```

```

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

```



```

3808
3809
3810
3811
3812
3813
3814
3815
3816 012634
3817 012634 042777 000001 166760
3818 012642 112767 000001 166444
3819 012650 032737 002000 177570
3820 012656 001402
3821 012660 104400 012056
3822 012664 005267 166422
3823 012670 001775
3824 012672 011667 166426
3825 012676 162767 000002 166420
3826 012704 117767 166414 166410
3827 012712 032737 020000 177570
3828 012720 001004
3829 012722 004737 012776
3830 012726 104400 012453
3831 012732 005737 177570
3832 012736 100001
3833 012740 000000
3834 012742 032737 001000 177570
3835 012750 001403
3836 012752 016716 166332
3837 012756 000002
3838 012760 005767 000010
3839 012764 001402
3840 012766 016716 000002
3841 012772 000002
3842 012774 000000
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852 012776 104400 012453
3853
3854 013002 010046
3855 013004 005000
3856 013006 116700 166310
3857 013012 005300
3858 013014 006300
3859 013016 006300
3860 013020 006300
3861 013022 062700 001500
3862 013026 012067 000002
3863 013032 104400

```

```

*****
; 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
$HLT:
      BIC      #BIT0, $PARITY
      MOV      #1, $ERFLG           ; SET THE ERROR FLAG
      BIT      #SW10, $SWR         ; BELL ON ERROR?
      BEQ      1$,                ; NO - SKIP
      TYPE     $BELL              ; RING BELL
      INC      $ERTTL             ; COUNT THE NUMBER OF ERRORS
      BEQ      1$,                ; INSURE $ERTTL NOT=0
      MOV      (SP), $HLTAD        ; GET ADDRESS OF HLT INSTRUCTION
      SUB      #2, $HLTAD
      MOV      $SWR, $ITEMB        ; STRIP AND SAVE THE ERROR ITEM CODE
      BIT      #SW13, $SWR        ; SKIP TYPEOUT IF SET
      BNE      2$,                ; SKIP TYPEOUTS
      JSR      PC, $TYPERR         ; GO TO USER ERROR ROUTINE
      TYPE     $SCLF
      TST      $SWR               ; HALT ON ERROR
      BPL      3$,                ; SKIP IF CONTINUE
      HALT
      BIT      #SW09, $SWR        ; HALT ON ERROR!
      BEQ      4$,                ; LOOP ON ERROR SWITCH SET?
      MOV      $LPERR, (SP)       ; BR IF NO
      RTI
      TST      $ESCAPE           ; FUDGE RETURN FOR LOOPING
      BEQ      5$,                ; CHECK FOR AN ESCAPE ADDRESS
      MOV      $ESCAPE, (SP)     ; BR IF NONE--TAKE NORMAL RETURN
      RTI
      $ESCAPE: 0                  ; FUDGE RETURN ADDRESS FOR ESCAPE
      ; 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     , $SCLF          ; TYPE A CARRIAGE RETURN
                                           ; AND LINE FEED
      MOV      RO, -(SP)         ; SAVE RO CONTENTS
      CLR      RO                ; CLEAR RO
      MOV      $ITEMB, RO        ; PICK UP THE ITEM INDEX
      DEC      RO                ; ADJUST THE INDEX
      ASL      RO                ; SO IT WILL WORK FOR
      ASL      RO                ; FOR THE ERROR TABLE
      ASL      RO
      ADD      # $ERTTB, RO      ; FORM THE TABLE POINTER
      MOV      (RO)+, 1$         ; PICK UP 'ERROR MESSAGE' POINTER
      TYPE

```



```

3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901 013124 010546
3902 013126 010605
3903 013130 012703 077406
3904 013134 010377 165050
3905 013140 010377 165046
3906 013144 010377 165044
3907 013150 010377 165042
3908 013154 010377 165040
3909 013160 010377 165036
3910 013164 010377 165034
3911 013170 010377 165032
3912 013174 005077 165030
3913 013200 012777 000200 165024
3914 013206 012777 000400 165020
3915 013214 012777 000600 165014
3916 013222 012777 001000 165010
3917 013230 012777 001200 165004
3918 013236 012777 001400 165000
3919 013244 012777 007600 164774
3920
3921
3922 013252 016704 164766
3923 013256 005014
3924 013260 005277 164720
3925 013264 010746
3926 013266 062716 000026
3927
3928 013272 012637 000004
3929 013276 005737 143776
3930 013302 062714 000040
3931 013306 027714 164734
3932 013312 003371
3933 013314 011400
3934 013316 162700 000040
3935 013322 012737 000006 000004
3936 013330 010506
3937 013332 010067 000010
3938 013336 005077 164642
3939 013342 012605
3940 013344 000207
3941 013346 000000
3942
3943
3944
3945
3946

```

```

*****
: THE FOLLOWING ROUTINE WILL SIZE MEMORY
: $LSTBLK WILL CONTAIN THE LAST BANK AS AN SAF (SEGMENT ADDRESS FIELD)
: THE SEGMENT ADDRESS FIELD CONTAINS THE 4 MOST SIGNIFICANT BITS OF
: THE LAST ADDRESS OF THE LAST BANK FOUND
*****
$SIZE: MOV R5, -(SP) ;SAVE R5 CONTENTS
MOV SP, R5 ;SAVE THE STACK POINTER
MOV #77406, R3
MOV R3, @KPDRO ;SET THE FOLLOWING PAGE
MOV R3, @KPDRI ;DESCRIPTOR REGISTERS TO
MOV R3, @KPDR2 ;READ/WRITE AND TRANSFER OF
MOV R3, @KPDR3 ;4096 (10) WORDS PER SEGMENT
MOV R3, @KPDR4
MOV R3, @KPDR5
MOV R3, @KPDR6
MOV R3, @KPDR7
CLR @KPAR0 ;SET THE FOLLOWING PAGE
MOV #200, @KPAR1 ;ADDRESS REGISTERS TO THEIR
MOV #400, @KPAR2 ;RESPECTIVE OFFSET VALUES
MOV #600, @KPAR3 ;FOR RELOCATION PURPOSES
MOV #1000, @KPAR4
MOV #1200, @KPAR5
MOV #1400, @KPAR6
MOV #7600, @KPAR7 ;THIS ONE'S THE I/O RECORD
;PAGE CONTAINING CONTROL STATUS
;REGISTERS, ETC.
MOV KPAR6, R4 ;GET ADDRESS OF PAGE 6 REGISTER
CLR (R4) ;CLEAR THE REGISTER
INC @SRO ;TURN ON MEMORY MANAGEMENT
MOV PC, -(SP) ;MAKE KTI1 TIMEOUT SERVICE
ADD #SKTOUT--, (SP) ;ROUTINE ADDRESS POSITION
;INDEPENDENT
MOV (SP)+, @ERRVEC ;SET FOR TIMEOUT
IS: TST @#143776 ;TRAP ON NON-EXISTENT MEMORY
ADD #40, (R4) ;MAKE A 1K STEP
CMP @KPAR7, (R4) ;LAST ONE?
BGT IS ;NO - TRY IT!
SKTOUT: MOV (R4), R0 ;GET LAST BANK +1
SUB #40, R0 ;DROP BACK
MOV #6, @ERRVEC ;SET FOR ERRORS
MOV R5, SP ;RESTORE THE STACK POINTER
MOV R0, $LSTBLK ;STORE THE SAF
CLR @SRO ;TURN MEMORY MGMT OFF
IS: MOV (SP)+, R5 ;RESTORE R5
RTS PC ;RETURN TO NORMAL FLOW
$LSTBLK: .WORD 0 ;CONTAINS THE SAF
*****
: BINARY TO OCTAL (ASCII) AND TYPE
: $B2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
: CALL:
MOV NUM, -(SP) ;NUMBER TO BE TYPED

```

```

3947      JSR      RO,$B2OCT      ;CALL FOR TYPEOUT
3948      .BYTE   N                ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3949      .BYTE   M                ;M=1 OR 0
3950
3951      ;1=TYPE LEADING ZEROS
3952      ;0=SUPPRESS LEADING ZEROS
3953
3954      $B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B2OCT OR $B2016
3955      CALL:
3956      MOV      NUM,-(SP)
3957      JSR      RO,$B201
3958
3959      $B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3960      CALL:
3961      MOV      NUM,-(SP)
3962      JSR      RO,$B2016
3963      013350 112067 000201      $B20CT: MOVB   (RO)+,$OMODE+1      ;PICKUP THE NUMBER OF DIGITS TO TYPE
3964      013354 112067 000173      MOVB   (RO)+,$OFILL      ;GET THE ZERO FILL SWITCH
3965      013360 000406      BR      $B201
3966      013362 112767 000001 000163 $B2016: MOVB   #1,$OFILL      ;SET THE ZERO FILL SWITCH
3967      013370 112767 000006 000157      MOVB   #6,$OMODE+1      ;SET FOR SIX(6) DIGITS
3968      013376 112767 000005 000146 $B201: MOVB   #5,$OCNT      ;SET THE ITERATION COUNT
3969      013404 010346      MOV      R3,-(SP)      ;SAVE R3
3970      013406 010446      MOV      R4,-(SP)      ;SAVE R4
3971      013410 010546      MOV      R5,-(SP)      ;SAVE R5
3972      013412 116704 000137      MOVB   $OMODE+1,R4      ;GET THE NUMBER OF DIGITS TO TYPE
3973      013416 005404      NEG      R4
3974      013420 062704 000006      ADD      #6,R4      ;SUBTRACT IT FOR MAX. ALLOWED
3975      013424 110467 000124      MOVB   R4,$OMODE      ;SAVE IT FOR USE
3976      013430 116704 000117      MOVB   $OFILL,R4      ;GET THE ZERO FILL SWITCH
3977      013434 016605 000010      MOV      10(SP),R5      ;PICKUP THE INPUT NUMBER
3978      013440 005003      CLR      R3      ;CLEAR THE OUTPUT WORD
3979      013442 006105      1$:   ROL      R5      ;ROTATE MSB INTO "C"
3980      013444 000404      BR      3$      ;GO DO MSB
3981      013446 006105      2$:   ROL      R5      ;FORM THIS DIGIT
3982      013450 006105      ROL      R5
3983      013452 006105      ROL      R5
3984      013454 010503      MOV      R5,R3
3985      013456 006103      3$:   ROL      R3      ;GET LSB OF THIS DIGIT
3986      013460 105367 000070      DECB   $OMODE      ;TYPE THIS DIGIT?
3987      013464 100016      BPL     7$      ;BR IF NO
3988      013466 042703 177770      BIC     #177770,R3      ;GET RID OF JUNK
3989      013472 001002      BNE     4$      ;TEST FOR 0
3990      013474 005704      TST     R4      ;SUPPRESS THIS 0?
3991      013476 001403      BEQ     5$      ;BR IF YES
3992      013500 005204      4$:   INC      R4      ;DON'T SUPPRESS ANYMORE 0'S
3993      013502 052703 000060      BIS     #'0,R3      ;MAKE THIS DIGIT ASCII
3994      013506 052703 000040      5$:   BIS     #' ,R3      ;MAKE ASCII IF NOT ALREADY
3995      013512 110367 000032      MOVB   R3,$S      ;SAVE FOR TYPING
3996      013516 104400 013550      TYPE   $S      ;GO TYPE THIS DIGIT
3997      013522 105367 000024      7$:   DECB   $OCNT      ;COUNT BY 1
3998      013526 003347      BGT     2$      ;BR IF MORE TO DO
3999      013530 002402      BLT     6$      ;BR IF DONE
4000      013532 005204      INC     R4      ;INSURE LAST DIGIT ISN'T A BLANK
4001      013534 000744      BR      2$      ;GO DO THE LAST DIGIT
4002      013536 012605      6$:   MOV     (SP)+,R5      ;RESTORE R5

```

K07

MAINDEC-11-DCKBR-E
DCKBRE.P11

MACY11 27(732) 16-SEP-76 16:25 PAGE 85
BINARY TO OCTAL (ASCII) AND TYPE

4003	013540	012604
4004	013542	012603
4005	013544	012616
4006	013546	000200
4007	013550	000
4008	013551	000
4009	013552	000
4010	013553	000
4011	013554	000000

	MOV	(SP)+,R4
	MOV	(SP)+,R3
	MOV	(SP)+,(SP)
	RTS	RO
BS:	.BYTE	0
	.BYTE	0
\$OCNT:	.BYTE	0
\$OFILL:	.BYTE	0
\$OMODE:		0

;RESTORE R4
;RESTORE R3
;SET THE STACK FOR RETURNING
;RETURN
;STORAGE FOR ASCII DIGIT
;TERMINATOR FOR TYPE ROUTINE
;OCTAL DIGIT COUNTER
;ZERO FILL SWITCH
;NUMBER OF DIGITS TO TYPE

```

4012
4013
4014
4015 013556 010046
4016 013560 016600 000002
4017 013564 005740
4018 013566 111000
4019 013570 016000 013576
4020 013574 000200
4021
4022
4023
4024
4025
4026 013576 001110
4027 013600 012456
4028 013602 012512
4029 013604 012314

```

```

;TRAP HANDLER
$TRAP:  MOV    RO, -(SP)      ;SAVE RO
        MOV    2(SP), RO    ;GET TRAP ADDRESS
        TST   -(RO)        ;BACKUP BY 2
        MOVB  (RO), RO      ;GET RIGHT BYTE OF TRAP
        MOV   $TRPAD(RO), RO ;INDEX TO TABLE
        RTS   RO           ;GO TO ROUTINE

```

```

;TRAP TABLE
;ROUTINE
;-----
$TRPAD: $TYPE  ;CALL=TYPE   TRAP+0(104400) TTY TYPEOUT ROUTINE
        $READC ;CALL=GETCHR TRAP+2(104402) TTY TYPEIN CHARACTER ROUTINE
        $READS ;CALL=GETSTR TRAP+4(104404) TTY TYPEIN STRING ROUTINE
        $ACCEPT ;CALL=ACCEPT TRAP+6(104406) READ AN OCTAL NUMBER FROM TTY

```

```

4030
4031 ;*****
4032 ;POWER DOWN ROUTINE
4033 013606 012737 013734 000024 $PWRDN: MOV $SILLUP, @#PWRVEC ;SET FOR FAST UP
4034 013614 012737 000340 000026 MOV #340, @#PWRVEC+2 ;PRIO:7
4035 013622 010046 MOV RO, -(SP) ;PUSH R0 ON STACK
4036 013624 010146 MOV R1, -(SP) ;PUSH R1 ON STACK
4037 013626 010246 MOV R2, -(SP) ;PUSH R2 ON STACK
4038 013630 010346 MOV R3, -(SP) ;PUSH R3 ON STACK
4039 013632 010446 MOV R4, -(SP) ;PUSH R4 ON STACK
4040 013634 010546 MOV R5, -(SP) ;PUSH R5 ON STACK
4041 013636 010667 000076 MOV SP, $$SAVR6 ;SAVE SP
4042 013642 012737 013654 000024 MOV $PWRUP, @#PWRVEC ;SET UP VECTOR
4043 013650 000000 HALT
4044 013652 000776 BR .-2 ;HANG UP
4045 ;*****
4046 ;POWER UP ROUTINE
4047 013654 016706 000060 $PWRUP: MOV $$SAVR6, SP ;GET SP
4048 013660 005067 000054 CLR $$SAVR6 ;WAIT LOOP FOR THE TTY
4049 013664 005267 000050 1$: INC $$SAVR6 ;WAIT FOR THE INC
4050 013670 001375 BNE 1$ ;OF WORD
4051 013672 012605 MOV (SP)+, R5 ;POP STACK INTO R5
4052 013674 012604 MOV (SP)+, R4 ;POP STACK INTO R4
4053 013676 012603 MOV (SP)+, R3 ;POP STACK INTO R3
4054 013700 012602 MOV (SP)+, R2 ;POP STACK INTO R2
4055 013702 012601 MOV (SP)+, R1 ;POP STACK INTO R1
4056 013704 012600 MOV (SP)+, R0 ;POP STACK INTO R0
4057 013706 012737 013606 000024 MOV $PWRDN, @#PWRVEC ;SET UP THE POWER DOWN VECTOR
4058 013714 012737 000340 000026 MOV #340, @#PWRVEC+2 ;PRIO:7
4059 013722 104400 013742 TYPE $POWER ;POWER FAIL MESSAGE
4060 013726 012716 001706 MOV $BEGIN, (SP) ;RESTART AT BEGIN
4061 013732 000002 RTI
4062 013734 000000 $SILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
4063 013736 000776 BR .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
4064 013740 000000 $$SAVR6: 0 ;PUT THE SP HERE
4065 013742 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
4066 013750 000122 .EVEN
4067

```

```

4068
4069
4070
4071
4072
4073
4074 013752 042524 052123 042040
4075 013760 042111 023516 020124
4076 013766 041101 051117 020124
4077 013774 000040
4078 013776 040506 040524 020114
4079 014004 051105 047522 020122
4080 014012 047524 050040 047522
4081 014020 051107 046501 020040
4082 014026 000
4083 014027 101 047502 052122
4084 014034 042105 044440 041516
4085 014042 051117 042522 052103
4086 014050 054514 020040 000
4087 014055 116 020117 040520
4088 014062 044522 054524 046440
4089 014070 046505 051117 020131
4090 014076 047506 047125 020104
4091 014104 042502 047514 020127
4092 014112 034062 020113 000040
4093 014120 042522 042523 020124
4094 014126 047504 051505 023516
4095 014134 020124 047527 045522
4096 014142 020040 000
4097 014145 125 042523 020122
4098 014152 042523 042514 052103
4099 014160 042105 051040 043505
4100 014166 051511 042524 020122
4101 014174 047516 020124 051120
4102 014202 051505 047105 020124
4103 014210 000040
4104 014212 047516 050040 051101
4105 014220 052111 020131 042515
4106 014226 047515 054522 043040
4107 014234 052517 042116 040440
4108 014242 020124 046101 020114
4109 014250 000040
4110 014252 044504 047104 052047
4111 014260 040440 047502 052122
4112 014266 047440 020122 042522
4113 014274 047503 047107 055111
4114 014302 020105 052123 041501
4115 014310 020113 044526 046117
4116 014316 052101 047511 020116
4117 014324 000040
4118 014326 041101 051117 042524
4119 014334 020104 052502 020124
4120 014342 052123 041501 020113
4121 014350 044526 046117 052101
4122 014356 047511 020116 047516
4123 014364 020124 042522 047503

```

```

:*****
:ERROR AND MESSAGE TABLE CONDIMENTS
:*****

```

```

EM1: .ASCIZ /TEST DIDN'T ABORT /
EM2: .ASCIZ /FATAL ERROR TO PROGRAM /
EM3: .ASCIZ /ABORTED INCORRECTLY /
EM4: .ASCIZ /NO PARITY MEMORY FOUND BELOW 28K /
EM5: .ASCIZ /RESET DOESN'T WORK /
EM6: .ASCIZ /USER SELECTED REGISTER NOT PRESENT /
EM7: .ASCIZ /NO PARITY MEMORY FOUND AT ALL /
EM10: .ASCIZ /DIDN'T ABORT OR RECOGNIZE STACK VIOLATION /
EM11: .ASCIZ /ABORTED BUT STACK VIOLATION NOT RECOGNIZED /

```


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

		3452*	3483	3485*	3606*	3607	3609*	3631*	3634*	3649*	3651	3654	3720	3724*
		3748*	3749	3753*	3854	3855*	3856*	3857*	3858*	3859*	3860*	3861*	3862	3867
		3873*	3875*	3880	3882*	3885	3933*	3934*	3937	3963	3964	4006*	4015	4016*
		4017	4018*	4019*	4020*	4035	4056*							
R1	=%000001	764*	1484*	1487*	1488	1584	1695	1711	1733	1755	1777	1799	1821	1843
		1865	1887	1890	1912	1934	1938	1959	1982	2004	2026	2029	2051	2055
		2076	2098	2120	2142	2164	2186	2190	2211	2233	2255	2494	2539	2570
		2754	2776	2798	2820	2842	2859	3222	3253	3284	3315	3365	3415	3578
		3721	3725*	3739*	3740*	3741*	3743*	3748	3752*	4036	4055*			
R2	=%000002	765*	1291*	1328*	1339	1342	1435	1443*	1447	1456*	1465*	1470*	1498	1573
		1618*	1619*	2335*	2337*	2340	2343*	2347*	2349*	2352*	2354	2358*	2365*	2368
		2431*	2433*	2454	2457*	2460*	2462*	2465*	2467	2494*	2509*	2515*	2520	2540*
		2543	2546*	3315*	3316*	3318*	3321*	3323*	3331	3336*	3339*	3341	3345*	3348
		3365*	3366*	3368*	3371*	3373*	3381	3386*	3389*	3391	3395*	3398	3415*	3416*
		3418*	3421*	3423*	3431	3436*	3439*	3441	3445*	3448	3546	3549*	3560*	3568*
		3575*	3587	3590	3596*	3612*	3616	3622*	3722	3726*	3737*	3751*	4037	4054*
R3	=%000003	766*	1289*	1290*	1367*	1489*	1569	1571*	1612	1693*	2495*	2498*	2502*	2505
		2520*	2570*	2615*	2619*	3236*	3267*	3298*	3348*	3398*	3448*	3472*	3491*	3503*
		3530*	3547	3573*	3578	3595*	3610*	3621*	3723	3728*	3729	3731	3733	3735
		3742*	3743	3750*	3781	3782*	3783	3786*	3787	3791	3793	3795*	3797*	3903*
		3904	3905	3906	3907	3908	3909	3910	3911	3969	3978*	3984*	3985*	3988*
		3993*	3994*	3995	4004*	4038	4053*							
R4	=%000004	767*	1331*	1353	1384*	1387	1389	1395*	1397	1416*	1419	1421	1427*	1429
		1608*	1620	3548	3550*	3581*	3586*	3587	3592	3594*	3618	3620*	3922*	3923*
		3930*	3931	3933	3970	3972*	3973*	3974*	3975	3976*	3990	3992*	4000*	4003*
		4039	4052*											
R5	=%000005	768*	1295*	1567	1581*	1582	1589*	1595*	1603*	1608*	1620*	1701*	1709	1731
		1753	1775	1797	1819	1841	1863	1885	1910	1932	1957	1980	2002	2024
		2049	2074	2096	2118	2140	2162	2184	2209	2231	2253	2315	2339	2356*
		2374*	2419*	2429	2452	2468*	2473*	2478*	2488	2537	2607	2665	2723	2752
		2774	2796	2818	2840	2927	2985	3055	3159	3219	3250	3281	3312	3362
		3412	3901	3902*	3936	3939*	3971	3977*	3979*	3981*	3982*	3983*	3984	4002*
		4040	4051*											
R6	=%000006	769*	771											
R7	=%000007	770*	772	1343*	1368*	1553*	1557*	1590*	1600*					
SAVLOC	010126	3064*	3101											
SEGVEC	000250	889*												
SP	=%000006	771*	922*	923	924*	926	927	928*	931	933*	935*	941	1208*	1254*
		1255*	1261	1264	1284	1287	1326*	1330*	1446	1459	1469	1567*	1568*	1588
		1589	1594	1595	1599*	1601	1602	1603	1604	1723*	1745*	1767*	1789*	1911*
		1833*	1855*	1877*	1902*	1924*	1949*	1972*	1994*	2016*	2041*	2066*	2088*	2110*
		2132*	2154*	2176*	2201*	2223*	2245*	2267*	2333*	2339*	2344*	2354*	2356	2359
		2378*	2452*	2458*	2467*	2468	2477*	2529*	2576*	2623*	2667*	2668*	2680*	2725*
		2726*	2734*	2741*	2766*	2788*	2810*	2832*	2854*	2945*	3009*	3061*	3073	3094*
		3095*	3100*	3164*	3176	3197*	3198*	3203*	3242*	3273*	3304*	3354*	3404*	3454*
		3546*	3547*	3548*	3594	3595	3596	3602	3605	3620	3621	3622	3627	3630
		3674*	3677	3679	3680	3706	3707	3710*	3720*	3721*	3722*	3723*	3724	3729
		3749*	3750	3751	3752	3753	3767*	3768*	3771*	3772*	3781*	3786	3797	3798*
		3799*	3800*	3824	3836*	3840*	3854*	3875	3880*	3901*	3902	3925*	3926*	3928
		3936*	3939	3969*	3970*	3971*	3977	4002	4003	4004	4005*	4015*	4016	4035*
		4036*	4037*	4038*	4039*	4040*	4041	4047*	4051	4052	4053	4054	4055	4056
		4060*												
SR0	000204	862*	1268	1269*	1282*	3924*	3938*							
SR2	000206	863*												
STACK	= 001100	753*	1208	1723	1745	1767	1789	1811	1833	1855	1877	1902	1924	1949
		1972	1994	2016	2041	2066	2088	2110	2132	2154	2176	2201	2223	2245

.SSNDU	40248	728
.STERM	40228	
.STRAD	40248	4012
.STYDE	40248	890

ADD	928	1358	1361	1364	1378	1384	1410	1416	1443	1456	1465	1599	1619	1688	1961
	23352	2365	2465	2498	2502	2759	2982	3464	3465	3467	3470	3610	3612	3861	3926
ASL	1272	3739	3740	3741	3858	3859	3860								
BFA	919	1316	1334	1377	1388	1404	1409	1420	1440	1453	1461	1513	1525	1538	1551
	1599	1607	1676	2342	2437	2456	2497	2545	2561	2872	2881	2591	3328	3378	3429
	3559	3604	3629	3650	3684	3686	3688	3690	3699	3732	3820	3823	3835	3839	3858
	3886	3991													
BCE	3702														
BCT	3734	3932	3998												
BCT	2865	3580													
BIC	1511	1523	1539	1587	1593	1717	1720	1739	1742	1761	1764	1783	1786	1805	1808
	1827	1830	1849	1852	1871	1874	1889	1896	1899	1918	1921	1935	1943	1946	1966
	1969	1988	1991	2010	2013	2028	2035	2038	2052	2060	2063	2082	2085	2104	2107
	2126	2129	2148	2151	2170	2173	2187	2195	2198	2217	2220	2239	2242	2261	2264
	2317	2355	2375	2435	2453	2474	2490	2514	2526	2542	2555	2573	2609	2618	2620
	2675	2677	2733	2738	2760	2763	2782	2785	2804	2807	2826	2829	2848	2851	2929
	2937	2942	3001	3006	3060	3086	3163	3189	3221	3229	3239	3252	3260	3270	3293
	3291	3301	3314	3344	3351	3364	3394	3401	3414	3444	3451	3459	3586	3742	3772
	3817	3988													
BIS	1503	1519	1535	1547	1891	1937	2030	2054	2189	2346	2451	2508	2551	2614	2736
	2935	3226	3257	3288	3326	3376	3426	3484	3743	3993	3994				
BISB	2847														
BISB	1314	1504	1512	1520	1524	1537	1550	3646	3683	3689	3696	3819	3827	3834	
BISB	3784														
BISB	936	3736	3738	3999											
BISB	1276	1532	3670												
BISB	925	932	1505	1521	2361	3089	3192	3462	3555	3589	3647	3652	3697	3788	3794
BISB	3828	3874	3989	4050											
BISB	940	1340	1357	1372	1686	3567	3770	3832	3987						
BISB	921	938	1259	1283	1319	1327	1363	1379	1393	1401	1411	1425	1433	1445	1677
	1719	1741	1763	1785	1907	1829	1851	1873	1898	1920	1945	1968	1990	2012	2037
	2062	2084	2106	2128	2150	2172	2197	2219	2241	2263	2345	2373	2447	2459	2472
	2501	2519	2524	2550	2572	2617	2672	2674	2730	2732	2762	2784	2806	2828	2850
	2874	2941	2997	3005	3080	3084	3092	3183	3187	3195	3238	3269	3300	3338	3350
	3388	3400	3438	3450	3564	3572	3615	3672	3678	3681	3692	3695	3744	3746	3790
	3888	3965	3980	4001	4044	4063									
CLR	1211	1218	1219	1220	1222	1223	1225	1228	1229	1230	1231	1232	1233	1234	1235
	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250
	1251	1253	1265	1269	1281	1288	1335	1391	1399	1423	1431	1470	1622	1623	3057
	3102	3103	3105	3161	3205	3206	3208	3217	3499	3501	3510	3512	3549	3550	3560
	3643	3644	3694	3708	3725	3855	3912	3923	3938	3978	4048				
CLPB	1221	2803	3693	3795											
CLPB	1261	1284	1339	1403	1446	1459	1486	1597	1601	1804	1826	1848	1870	1895	1917
	1942	1965	1987	2009	2034	2059	2081	2103	2125	2147	2169	2194	2859	2967	2979
	3578	3587	3602	3627	3651	3679	3701	3731	3733	3735	3783	3931			
CLPB	931	3685	3787	3793											
DEC	1277	1373	1405	2337	2343	2349	2433	2457	2462	2546	3737	3857			
DECB	935	3986	3997												
EMT	755														
HALT	848	920	3833	4043	4062										
INC	1262	1282	1325	1554	1596	2886	3645	3700	3705	3822	3924	3992	4000	4049	
ICT	756														
JMP	851	1341	1441	1454	1458	1464	1467	1472	1610	1624	1678	2216	2238	2260	2985
	3473	3492	3658												
JSR	930	937	1280	1289	1343	1368	1571	1582	1693	1699	1709	1721	1731	1743	1753

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

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

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

