

KK11-B

11/44 KK11B CACHE  
CKKKABO

AH-F626B-MC  
FICHE 1 OF 2

FEB 1981  
COPYRIGHT © 79-80  
MADE IN USA



A large grid of approximately 15 columns and 20 rows of small, dense data tables. Each cell in the grid contains a small table with multiple columns and rows of text, likely representing individual data points or records. The text is too small to be legible, but the overall structure is a comprehensive data matrix.

KK11-B

11/44 KK11B CACHE  
CKKKABO

AH-F626B-MC  
FICHE 2 OF 2

FEB 1981  
COPYRIGHT © 79-80  
MADE IN USA



.REM %

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

IDENTIFICATION

PRODUCT CODE:	AC-F624B-MC
PRODUCT NAME:	CKKKABO 11/44 KK11B CACHE
DATE CREATED:	OCTOBER 1980
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	JOHN CIUKAJ

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1979,1980 BY DIGITAL EQUIPMENT CORPORATION

44  
45  
46  
47  
48  
49  
50

1.0 HISTORY SECTION  
-----

CKKKAAO WAS RELEASED OCT 1979  
CKKKABO WAS RELEASED OCT 1980

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

TABLE OF CONTENTS  
-----

- 1.0 HISTORY SECTION
- 2.0 GENERAL PROGRAM INFO
  - 2.1 ABSTRACT
  - 2.2 TEST STRUCTURE
  - 2.3 HARDWARE REQUIREMENTS
    - 2.3.1 REQUIRED EQUIPMENT
    - 2.3.2 OPTIONAL EQUIPMENT
    - 2.3.3 DIAGNOSTIC PREREQUISITES
    - 2.3.4 RELATED DOCUMENTS
- 3.0 OPERATING INSTRUCTIONS
  - 3.1 LOAD AND START PROCEDURE
  - 3.2 SWITCH REGISTER OPTIONS
    - 3.2.1 OPTIONS
    - 3.2.2 LOOP ON ERROR
    - 3.2.3 LOOP ON TESTS
    - 3.2.4 IMPLEMENTATION
  - 3.3 APT
    - 3.3.1 USER SWITCH REGISTER
    - 3.3.2 PROGRAM LOAD FILE
  - 3.4. EXECUTION TIMES
- 4.0 ERROR INFO
  - 4.1 ERROR PRINTOUTS
  - 4.2 UNCONTROLLED ERRORS
- 5.0 HANDLERS AND COMMON ROUTINES

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  
131  
132  
133  
134  
135  
136

2.0 GENERAL PROGRAM INFO  
-----  
2.1 ABSTRACT

THIS DIAGNOSTIC IS A LOGIC TEST OF THE 11/44 CACHE .  
IT IS APT,ACT11,AND XXDP COMPATIBLE.  
THIS DIAGNOSTIC ASSOCIATES A GROUP OF TESTS WITH ONE AREA  
OF CACHE LOGIC AND PROCEEDS TO TEST THAT AREA COMPREHENSIVELY.  
THE MAINTENANCE FEATURES OFFERED BY THE 11/44 CACHE ALLOWS  
INFORMATION TO BE READ IN KEY AREAS OF THE CACHE ALLOWING  
THE DIAGNOSTIC TO ISOLATE FAILURES TO DATA PATHS ,AND IN SOME  
CASES IC'S.

AT THE START OF THE DIAGNOSTIC, A SMALL AREA OF WRITE CONTROL  
LOGIC AND THE MAINTENANCE FEATURES ARE ASSUMED TO BE WORKING.  
SO EFFECTIVE ARE THE MAINTENANCE FEATURES THAT THE CACHE IS  
COMPLETELY TURNED OFF (NO DATA IS ALLOWED TO BE CACHED OUT OF  
THE CACHE) AT THE START OF THE DIAGNOSTIC AND NOT TURNED ON  
UNTIL 90 PERCENT OF THE DIAGNOSTIC IS COMPLETE.

TYPICAL TEST SEQUENCE FOR A BLOCK OF LOGIC CONTAINING  
RAM IC'S IS TO FIRST VERIFY DATA PATHS TO ONE RAM LOCATION,  
VERIFY THAT 0'S AND 1'S CAN BE WRITTEN TO ALL RAM LOCATIONS  
,VERIFY ADDRESS LINES TO RAMS, AND FINALLY TO CHECK THE  
INTEGRITY OF THE RAMS BY PERFORMING  
A MARCH PATTERN TEST.

THE DIAGNOSTIC TESTS WERE DESIGNED IN ASSOCIATION WITH  
A M7097 CACHE LOGIC SCHEMATIC. REFERENCE TO THIS DOCUMENT  
WILL HELP THE UNDERSTANDING OF THE TEST SEQUENCING AND PURPOSE.

UPON START OF THE PROGRAM, THE CACHE IS IMMEDIATELY TURNED OFF  
(FORCE MISS IS ON FOR BOTH HALVES OF CACHE, INTERRUPTS ARE DISABLED  
AND CACHE IS IN BYPASS MODE). THE TESTS THEN PROCEED TO SELECTIVELY  
TURN ON ONLY THE HALF OF CACHE THAT IS TO BE EXERCISED.  
THIS IS TO ENSURE THAT THE INSTRUCTIONS ARE NOT EXECUTED OUT  
OF A POSSIBLY BAD CACHE. IN ORDER TO IMPLEMENT THIS SCHEME,  
THE TESTS THAT ENABLE CACHE ARE RELOCATED TO AREAS OF CACHE  
THAT ARE NOT ENABLED. THE TESTS ARE STRUCTURED ON A HALF CACHE  
BASIS. THAT IS A TEST MAY BE RUN IN LOW CACHE WHILE TESTING  
HIGH CACHE AFTER WHICH AN IDENTICAL TEST WILL RUN IN HIGH CACHE  
WHILE TESTING LOW CACHE.

TO FACILITATE THE TESTING OF CACHE, A 4K BUFFER IS RESERVED AT THE  
END OF THE PROGRAM FOR READ WRITE OPERATIONS AND RELOCATION OF TESTS.

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  
187  
188  
189  
190  
191

2.2 TEST STRUCTURE

EACH TEST IS STRUCTURED WITH THE FOLLOWING DEDICATED LOCAL SYMBOLS:

- 40\$: LOCATION WHERE TEST BEGINS
- 1\$: LOCATION OF THE BEGINNING OF THE LOOP ON ERROR CODE LOOP
- 25\$: LOCATION OF THE END OF THE LOOP ON ERROR CODE LOOP
- 10\$: LOCATION WHERE TEST ENDS

THESE LOCATIONS ARE USED BY THE \$SCPSET ROUTINE TO SET UP LOOP ON TEST AND LOOP ON ERROR VECTORS (REFER TO \$SCPSET SECT. 9.0)

2.3 HARDWARE REQUIREMENTS

2.3.1 REQUIRED EQUIPMENT

- 1. PDP11-44 CPU
  - A. M7094/M7095 CPU CONTROL DATA PATH
  - B. M7096 MFM
  - C. M7098 UBI
  - D. M7090 CIM
- 2. 16K MEMORY
- 3. I/O TERMINAL

2.3.2 OPTIONAL EQUIPMENT

- 1. RMI REGISTER(G5179) HARDWARE FOR HI ORDER ADDRESS LINE TESTING
- 2. PDP11 CPU UNIBUS EXERCISER

2.3.3 DIAGNOSTIC PREREQUISITES

IT IS ASSUMED THAT ALL THE ABOVE HARDWARE IS OPERATIONAL AND THAT THERE RESPECTIVE DIAGNOSTICS HAVE BEEN RUN FOR VERIFICATION.

2.3.4 RELATED DOCUMENTS

- 1. 11/44 CACHE DESIGN SPECIFICATION
- 2. M7097 CACHE LOGIC SCHEMATIC
- 3. RMI(G5179) REGISTER DESCRIPTION  
DATED 29 JAN 1979-PDP11 SYS. PROD. SUPPORT
- 4. PMK05 UNIBUS EXERCISER OPERATING AND SERVICE MANUAL
- 5. CFKKA 11/34 DIAGNOSTIC

A

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  
237  
238  
239  
240  
241  
242  
243

3.0 OPERATING INSTRUCTIONS

3.1 LOAD AND START PROCEDURE

1. LOAD PROGRAM INTO MEMORY
2. LOAD STARTING ADDRESS 200
3. START

LOADING AND STARTING AT 200 IS NORMAL LOGIC TESTING. THE FIRST PASS IS A QUICK VERIFY PASS FOLLOWED BY AN ENDPASS PRINTOUT. SUBSEQUENT EXECUTION OF THE PROGRAM WILL RESULT IN REPEATED PASSES SPECIFIED BY LOCATION \$TIMES BEFORE ENDPASS IS PRINTED AGAIN. ALL ERRORS ARE ACCOMPANIED BY AN ERROR PRINTOUT CONSISTING OF A MINIMUM OF THE FAILING TEST (TESTNO) AND THE LOCATION IN THE PROGRAM WHERE THE ERROR OCCURED (ERRPC). IT IS NECESSARY FOR THE USER TO REFER TO THE ASSEMBLED LISTING AT THE LOCATION SPECIFIED BY ERRPC FOR AN EXPLANATION OF THE ERROR.

3.2 SWITCH REGISTER OPTIONS

3.2.1 [OPTIONS]

SWITCH	OCTAL	FUNCTION
SW15=1	100000	HALT ON ERROR
SW14=1	040000	LOOP ON TEST SPECIFIED IN SW07:SW00
SW13=1	020000	INHIBIT ERROR TYPEOUTS
SW11=1	004000	INHIBIT ITERATIONS
SW09=1	001000	LOOP ON ERROR
SW08=1	000400	DIAGNOSTIC WILL TEST TO VERIFY THAT INVALIDATION WILL OCCUR DUE TO A READ HIT BYPASS CONDITION: DIAGNOSTIC ASSUMES PHYSICAL STRAP W2 IS IN. IF SW08=0 THEN DIAGNOSTIC TESTS TO VERIFY THAT NO INVALIDATION WILL OCCUR DUE TO A READ HIT BYPASS CONDITION. DIAGNOSTIC, IN THIS CASE, ASSUMES PHYSICAL STRAP W1 IS IN PLACE.
SW07 TO SW00	001-377	SPECIFIES TEST WHEN LOOP ON TEST IS SELECTED(SW14)



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  
291

### 3.2.2 LOOP ON ERROR

THE INTENT OF THE LOOP ON ERROR FEATURE (SW09) IS TO GET A TIGHT CODING LOOP TO OCCUR WHEN AN ERROR HAPPENS TO AID IN ISOLATING THE FAILURE. THE FOLLOWING IS A DESCRIPTION OF HOW THE PROGRAM HANDLES LOOP ON ERROR. FOR THIS EXAMPLE ASSUME THAT THE TEST HAS BEEN RELOCATED TO HI CACHE BUFFER AREA STARTING AT ADDRESS 70000.

1. AN ERROR OCCURS SO \$ERROR ROUTINE IS ENTERED
2. THE APPROPRIATE ERROR MESSAGE IS PRINTED
3. AN APPROPRIATE 'JMP 1\$' INSTRUCTION IS AUTOMATICALLY WRITTEN BY THE PROGRAM TO THE LOCATION IN HI CACHE BUFFER AREA LOCATION SPECIFIED BY 25\$ FOR THIS TEST.
4. THE \$ERROR ROUTINE WILL THEN JUMP TO THE LOCATION IN HI CACHE BUFFER AREA SPECIFIED BY 1\$ FOR THIS TEST.
5. THE PROGRAM WILL NOW BE EXECUTING A CODE LOOP IN HI CACHE BUFFER AREA BOUNDED BY THE LOCATIONS SPECIFIED BY 1\$ AND 25\$.

TO CLEAR THIS CONDITION THE CPU MUST BE HALTED FOLLOWED BY LOADING ADDRESS 200 AND START. IF SW09 BIT IS CLEARED THEN NORMAL PROGRAM EXECUTION WILL HAVE BEEN RESUMED.

### 3.2.3 LOOP ON TEST

WHEN LOOP ON TEST IS SELECTED (SW14) THE TEST SPECIFIED BY BITS 7:0 IN THE SWITCH REGISTER IS EXECUTED REPEATEDLY. THE TEST IS LOOPED IN ITS ENTIRETY, UNLIKE THE LOOP ON ERROR FEATURE.

### 3.2.4 IMPLEMENTATION

SELECT SWITCH REGISTER OPTIONS BY USING 11/44 MFM CONSOLE. TYPE ^P TO ENTER CONSOLE. NORMAL OPERATION IS TO RUN WITH ALL SWITCH REGISTER BITS EQUAL TO 0.

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  
344

3.3 APT  
---

3.3.1 THE FOLLOWING APT USER SWITCH REGISTER BITS ARE DEFINED FOR THIS DIAGNOSTIC AND ARE VALID ONLY IF \$ENVN BIT 7=1:

BIT 12 \$USWR ( UNIBUS EXERCISER)

- =1 APT SAYS PDP11 UNIBUS EXERCISER IS PRESENT SO PERFORM DMA TESTS.
- =0 APT SAYS DO NOT PERFORM DMA TESTS.

BIT 7 \$USWR ( RMI REGISTER (G5179))

- =1 APT SAYS RMI REGISTER IS PRESENT PERFORM HI ORDER ADDRESS LINE TESTS
- =0 APT SAYS DO NOT PERFORM HI ORDER ADDR. LINE TESTS.

3.3.2 THE FOLLOWING IS A PROGRAM LOAD FILE USED BY APT. E TABLE 'A' IS USED FOR APT DUMP MODE AND E TABLE 'B' IS USED FOR APT QV AND RUN TIME MODE. E TABLE 'B' IS SET UP TO RUN RMI REGISTER TESTS AND UNIBUS EXERCISER TESTS,INHIBIT ITERATIONS, AND SUPPRESS ERROR TYPEOUTS.

1ST PASS RUN TIME	LONGEST TEST TIME	ADDITIONAL RUN TIME
10	5	0

..... E TABLES .....

	A	B
E-MODE/S-MODE	200/000	240/001
SWITCH REGISTER 1	004000	004000
SWITCH REGISTER 2	010200	010200
CPU TYPE/OPTIONS	00/0000	00/0000
MEMORY MAP CODE 1	000/00000000	000/00000000
MEMORY MAP CODE 2	000/00000000	000/00000000
MEMORY MAP CODE 3	000/00000000	000/00000000
MEMORY MAP CODE 4	000/00000000	000/00000000
BUS PRIORITY/INTERRUPT 1	0000	0000
BUS PRIORITY/INTERRUPT 2	0000	0000
BUS ADDRESS CODE	000000	000000
DEVICE MAP CODE	000000	000000
CTLR. SPECIFIC WORD 1	000000	000000
CTLR. SPECIFIC WORD 2	000000	000000

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

3.4 EXECUTION TIMES

1ST PASS: LESS THAN 10 SEC.  
PASSES WITH ITERATIONS: LESS THAN 75 SEC.

4.0 ERROR INFO

4.1 IN ADDITION TO TESTNO AND ERRPC BEING PRINTED WHEN AN ERROR OCCURS, ADDITIONAL INFORMATION CAN BE GIVEN DEPENDING ON THE TEST. THE INFO. IS IN THE FORM OF DATA DESCRIBED IN A FASHION WHICH RELATES TO THE LOGIC BEING TESTED AND CAN AID IN ISOLATING THE FAILURE. FOR EXAMPLE, A TEST MAY VERIFY THAT THE CACHE TAG STORE RAMS CAN BE LOADED FROM THE CACHE ADDRESS LINES (CA<21:13>) AND THEN BE READ FROM THE CACHE HIT REGISTER BITS 15:7 (CHR<15:7>). AN ERROR PRINTOUT ,THEREFORE,WOULD LOOK LIKE THE FOLLOWING:

TESTNO	ERRPC	CHR157	CA2113
-----	-----	-----	-----
102	13234	001	000

WHERE: CHR157 SPECIFIES DATA READ FROM CACHE HIT REGISTER BITS 15:7

AND CA3113 SPECIFIES THE ADDRESS PATTERN ON THE CACHE ADDRESS LINES 21:13 USED TO LOAD THE TAG STORE

CA2113 IS ANALAGOUS TO 'DATA EXPECTED' AND CHR157 IS ANALAGOUS TO 'DATA RECEIVED'.

4.2 UNCONTROLLED ERRORS

IF AT ANY TIME THE PROGRAM STOPS WITHOUT PROPER ERROR INDICATION EXAMINING LOCATION SPECIFIED BY \$TESTN WILL INDICATE WHAT TEST THE PROGRAM HAD REACHED.

386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418

### S.O HANDLERS AND COMMON ROUTINES

-----

RELCTL: THIS ROUTINE, WHEN CALLED, WILL RELOCATE ALL TEST CODE OF THE TEST UP TO AND INCLUDING THE LOCATION SPECIFIED BY 10\$: TO LOW CACHE BUFFER AREA BEGINNING AT LOCATION 60000. WHEN THIS HAS BEEN DONE THE ROUTINE WILL JUMP TO LOCATION 60000 FOR TEST EXECUTION.

RELCTH: THIS ROUTINE, WHEN CALLED, WILL RELOCATE ALL TEST CODE OF THE TEST UP TO AND INCLUDING THE LOCATION SPECIFIED BY 10\$: TO HIGH CACHE BUFFER AREA BEGINNING AT LOCATION 70000. WHEN THIS HAS BEEN DONE THE ROUTINE WILL JUMP TO LOCATION 70000 FOR TEST EXECUTION.

\$SCPSET: THIS ROUTINE IS PERFORMED AT THE BEGINNING OF EACH TEST. IT SETS UP VECTORS TO ACCOMPLISH LOOP ON TEST AND LOOP ON ERROR. THE LOCATIONS SPECIFIED BY 40\$, 1\$, AND 25\$ ARE PASSED TO THE ROUTINE AND ARE ADDRESS LOCATIONS WHICH ARE INDICATIVE OF WHERE THOSE LOCATIONS WILL BE WHEN THE TEST IS RELOCATED TO EITHER HI OR LO CACHE BUFFER AREA.

\$ERROR: THIS ROUTINE IS CALLED WHEN THERE IS AN ERROR. IT WILL ALWAYS TYPE FAILING TEST NUMBER AND FAILING ERROR PC. IT MAY TYPE ADDITIONAL DATA INFO. DEPENDING ON THE TEST (USES THE ARGUMENTS PASSED BY THE TEST TO THE ROUTINE).

419  
512

```

%
.TITLE CKKKABO 11-44 KK11B CACHE
;*COPYRIGHT (C) OCT 1979
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY DAN P. MILLEVILLE
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

```

```

000001
160000
513 000000

000174
000174 000000
000176 000000

000200 000137 000200
520 000020
521 000020 002144
522 000022 000340
523 000030
524 000030 045246
525 000032 000340
526 000034 045172
527 000036 000340
528 000042
529 000042 000000
530 000046
531 000046 044146
532 000052
533 000052 000000

```

```

$TN=1
$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
.SBTTL TRAP CATCHER
.=0
;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
.=174
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
.SBTTL STARTING ADDRESS(ES)
JMP @#200 ;;JUMP TO STARTING ADDRESS OF PROGRAM
.=20
.WORD $SCPSET
.WORD 340
.=30
.WORD $ERROR
.WORD 340
.WORD $TRAP
.WORD 340
.=42
.WORD 0
.=46
.WORD $ENDAD
.=52
.WORD 0

```

```

534      000200      000200      . =200
535 000200 000137 001000      JMP      START
536      001000      001000      . =1000
537 001000 000005      START:   RESET      ;DISABLE ALL INTERRUPTS
538 001002 012706 000500      MOV      #500,SP      ;SET STACK POINTER
539 001006 012737 001032 000004      MOV      #48,4      ;SETUP FOR POSSIBLE NEX MEMORY TRAP
540 001014 012737 000340 000006      MOV      #340,6
541 001022 012737 001015 177746      MOV      #OFF,CCR      ;DISABLE CACHE
542 001030 000536      BR      5$      ;NO TRAP;CONTINUE
543 001032 022626      4$:      CMP      (SP)+,(SP)+      ;ADJUST STACK DUE TO TRAP
544 001034 012737 000006 000004      MOV      #6,4      ;RESTORE TRAP VECTORS
545 001042 005037 000006      CLR      6
546
547 001046 104401 001054      TYPE     ,65$      ;;TYPE ASCIZ STRING
      001052 000416      BR      64$      ;;GET OVER THE ASCIZ
      001054      200      103      113 65$: .ASCIZ <CRLF>/CKKKABO 11-44 KK11B CACHE/
      .EVEN
      64$:
548 001110 104401 001116      TYPE     ,67$      ;;TYPE ASCIZ STRING
      001114 000423      BR      66$      ;;GET OVER THE ASCIZ
      001116      200      124      122 67$: .ASCIZ <CRLF>/TRAP THRU NEX MEMORY VECTOR OCCURED/
      .EVEN
      66$:
549 001164 104401 001172      TYPE     ,69$      ;;TYPE ASCIZ STRING
      001170 000424      BR      68$      ;;GET OVER THE ASCIZ
      001172      200      104      111 69$: .ASCIZ <CRLF>/DIAGNOSTIC ATTEMPTED TO TURN CACHE OFF/
      .EVEN
      68$:
550 001242 104401 001250      TYPE     ,71$      ;;TYPE ASCIZ STRING
      001246 000423      BR      70$      ;;GET OVER THE ASCIZ
      001250      200      102      131 71$: .ASCIZ <CRLF>/BY ADDRESSING CACHE CONTROL REGISTER/
      .EVEN
      70$:
551 001316 012737 000001 001466      MOV      #1,$MSGTY      ;SET $MSGTY FOR POSSIBLE APT USE
552 001324 000000      HALT      ;HALT PROGRAM
553
554 001326 012737 000006 000004 5$:      MOV      #6,4      ;RESTORE VECTORS
555 001334 005037 000006      CLR      6
556 001340 005037 001474      CLR      $PASS      ;CLEAR PASS COUNT
557 001344 132737 000200 001507      BITB     #APTSIZE,$ENVM ;IS APT SIZING?
558 001352 001403      BEQ      1$      ;NO
559 001354 012737 001510 002074      MOV      #$$SWREG,$SWR ;YES;USE APT SWITCH REGISTER
560 001362 005737 000042 1$:      TST      42      ;IS THIS MANUAL MODE?
561 001366 001404      BEQ      3$      ;YES TYPE ID
562 001370 023737 000042 000046      CMP      42,46      ;IS THIS ACT 11 QV OR AUTO MODE?
563 001376 001423      BEQ      2$      ;YES;SKIP TITLE
564 001400 000240      3$:      NOP
565 001402 000240      NOP
566 001404 104401 001412      TYPE     ,73$      ;;TYPE ASCIZ STRING
      001410 000416      BR      72$      ;;GET OVER THE ASCIZ
      001412      200      103      113 73$: .ASCIZ <CRLF>/CKKKABO 11-44 KK11B CACHE /
      .EVEN
      72$:
567 001446      72$:
568 001446 000137 002416 2$:      JMP      BEGIN      ;START TEST

```

569

000024 001452  
000024 000024  
000024 000200  
000044 000044  
000044 001452  
001452 001452  
  
001452  
001452 000000  
001454 001466  
001456 000005  
001460 000010  
001462 000000  
001464 000052

```
.SBTTL APT PARAMETER BLOCK
:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
      .SX=.      ;;SAVE CURRENT LOCATION
      =24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
      200      ;;FOR APT START UP
      =44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
      $APTHDR  ;;POINT TO APT HEADER BLOCK
      =.SX     ;;RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM:  .WORD 5      ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 10     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 0      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

571

```
.SBTTL APT MAILBOX-ETABLE
:*****
.EVEN
001466 $MAIL:                ;; APT MAILBOX
001466 000000 $MSGTY: .WORD  AMSGTY  ;; MESSAGE TYPE CODE
001470 000000 $FATAL: .WORD  AFATAL  ;; FATAL ERROR NUMBER
001472 000000 $TESTN: .WORD  ATESTN  ;; TEST NUMBER
001474 000000 $PASS:  .WORD  APASS   ;; PASS COUNT
001476 000000 $DEVCT: .WORD  ADEVCT  ;; DEVICE COUNT
001500 000000 $UNIT:  .WORD  AUNIT   ;; I/O UNIT NUMBER
001502 000000 $MSGAD: .WORD  AMSGAD  ;; MESSAGE ADDRESS
001504 000000 $MSGLG: .WORD  AMSGLG  ;; MESSAGE LENGTH
001506 $ETABLE:                ;; APT ENVIRONMENT TABLE
001506 000 $ENV:  .BYTE  AENV   ;; ENVIRONMENT BYTE
001507 000 $ENVM:  .BYTE  AENVM  ;; ENVIRONMENT MODE BITS
001510 000000 $SWREG: .WORD  ASWREG  ;; APT SWITCH REGISTER
001512 000000 $USWR:  .WORD  AUSWR   ;; USER SWITCHES
001514 000000 $CPUOP: .WORD  ACPUOP  ;; CPU TYPE, OPTIONS
:                ;; BITS 15-11=CPU TYPE
:                ;; 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
:                ;; 11/70=06,PDQ=07,Q=10
:                ;; BIT 10=REAL TIME CLOCK
:                ;; BIT 9=FLOATING POINT PROCESSOR
:                ;; BIT 8=MEMORY MANAGEMENT
001516 000 $MAMS1: .BYTE  AMAMS1  ;; HIGH ADDRESS, M.S. BYTE
001517 000 $MTYP1: .BYTE  AMTYP1  ;; MEM. TYPE, BLK#1
:                ;; MEM. TYPE BYTE -- (HIGH BYTE)
:                ;; 900 NSEC CORE=001
:                ;; 300 NSEC BIPOLAR=002
:                ;; 500 NSEC MOS=003
001520 000000 $MADR1: .WORD  AMADR1  ;; HIGH ADDRESS, BLK#1
:                ;; MEM.LAST ADDR.=3 BYTES, THIS WORD AND LOW OF 'TYPE' ABOVE
001522 000 $MAMS2: .BYTE  AMAMS2  ;; HIGH ADDRESS, M.S. BYTE
001523 000 $MTYP2: .BYTE  AMTYP2  ;; MEM. TYPE, BLK#2
001524 000000 $MADR2: .WORD  AMADR2  ;; MEM.LAST ADDRESS, BLK#2
001526 000 $MAMS3: .BYTE  AMAMS3  ;; HIGH ADDRESS, M.S. BYTE
001527 000 $MTYP3: .BYTE  AMTYP3  ;; MEM. TYPE, BLK#3
001530 000000 $MADR3: .WORD  AMADR3  ;; MEM.LAST ADDRESS, BLK#3
001532 000 $MAMS4: .BYTE  AMAMS4  ;; HIGH ADDRESS, M.S. BYTE
001533 000 $MTYP4: .BYTE  AMTYP4  ;; MEM. TYPE, BLK#4
001534 000000 $MADR4: .WORD  AMADR4  ;; MEM.LAST ADDRESS, BLK#4
001536 000000 $VECT1: .WORD  AVECT1  ;; INTERRUPT VECTOR#1, BUS PRIORITY#1
001540 000000 $VECT2: .WORD  AVECT2  ;; INTERRUPT VECTOR#2, BUS PRIORITY#2
001542 000000 $BASE:  .WORD  ABASE   ;; BASE ADDRESS OF EQUIPMENT UNDER TEST
001544 000000 $DEVN:  .WORD  ADEVN   ;; DEVICE MAP
001546 000000 $CDW1:  .WORD  ACDW1   ;; CONTROLLER DESCRIPTION WORD#1
001550 000000 $CDW2:  .WORD  ACDW2   ;; CONTROLLER DESCRIPTION WORD#2
001552 000000 $DDW0:  .WORD  ADDW0   ;; DEVICE DESCRIPTOR WORD#0
001554 000000 $DDW1:  .WORD  ADDW1   ;; DEVICE DESCRIPTOR WORD#1
001556 000000 $DDW2:  .WORD  ADDW2   ;; DEVICE DESCRIPTOR WORD#2
001560 000000 $DDW3:  .WORD  ADDW3   ;; DEVICE DESCRIPTOR WORD#3
001562 000000 $DDW4:  .WORD  ADDW4   ;; DEVICE DESCRIPTOR WORD#4
001564 000000 $DDW5:  .WORD  ADDW5   ;; DEVICE DESCRIPTOR WORD#5
001566 000000 $DDW6:  .WORD  ADDW6   ;; DEVICE DESCRIPTOR WORD#6
001570 000000 $DDW7:  .WORD  ADDW7   ;; DEVICE DESCRIPTOR WORD#7
```



CKKKABO 11-44 KK11B CACHE  
APT MAILBOX-ETABLE

MACRO M1113 27-OCT-80 08:39 PAGE 15-1<sup>C 2</sup> SEQUENCE 15

001572 000000  
001574 000000  
001576 000000  
001600 000000  
001602 000000  
001604 000000  
001606 000000  
001610 000000  
001612

\$DDW8: .WORD ADDW8 :::DEVICE DESCRIPTOR WORD#8  
\$DDW9: .WORD ADDW9 :::DEVICE DESCRIPTOR WORD#9  
\$DDW10: .WORD ADDW10 :::DEVICE DESCRIPTOR WORD#10  
\$DDW11: .WORD ADDW11 :::DEVICE DESCRIPTOR WORD#11  
\$DDW12: .WORD ADDW12 :::DEVICE DESCRIPTOR WORD#12  
\$DDW13: .WORD ADDW13 :::DEVICE DESCRIPTOR WORD#13  
\$DDW14: .WORD ADDW14 :::DEVICE DESCRIPTOR WORD#14  
\$DDW15: .WORD ADDW15 :::DEVICE DESCRIPTOR WORD#15  
\$ETEND:

573

```

.SBTTL APT COMMUNICATIONS ROUTINE
*****
001612 112737 000001 002056 $ATY1: MOV  #1,$FFLG      ;;TO REPORT FATAL ERROR
001620 112737 000001 002054 $ATY3: MOV  #1,$MFLG     ;;TO TYPE A MESSAGE
001626 000403
001630 112737 000001 002056 $ATY4: MOV  #1,$FFLG     ;;TO ONLY REPORT FATAL ERROR
001636 $ATYC:
001636 010046      MOV  R0,-(SP)      ;;PUSH R0 ON STACK
001640 010146      MOV  R1,-(SP)      ;;PUSH R1 ON STACK
001642 105737 002054      TSTB $MFLG        ;;SHOULD TYPE A MESSAGE?
001646 001450      BEQ  5$           ;;IF NOT: BR
001650 122737 000001 001506      CMPB #APTENV,$ENV   ;;OPERATING UNDER APT?
001656 001031      BNE  3$           ;;IF NOT: BR
001660 132737 000100 001507      BITB #APTPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
001666 001425      BEQ  3$           ;;IF NOT: BR
001670 017600 000004      MOV  @4(SP),R0     ;;GET MESSAGE ADDR.
001674 062766 000002 000004      ADD  #2,4(SP)      ;;BUMP RETURN ADDR.
001702 005737 001466      1$: TST  $MSGTYPE    ;;SEE IF DONE W/ LAST XMISSION?
001706 001375      BNE  1$           ;;IF NOT: WAIT
001710 010037 001502      MOV  R0,$MSGAD     ;;PUT ADDR IN MAILBOX
001714 105720      2$: TSTB (R0)+      ;;FIND END OF MESSAGE
001716 001376      BNE  2$
001720 163700 001502      SUB  $MSGAD,R0     ;;SUB START OF MESSAGE
001724 006200      ASR  R0            ;;GET MESSAGE LNTH IN WORDS
001726 010037 001504      MOV  R0,$MSGGLT   ;;PUT LENGTH IN MAILBOX
001732 012737 000004 001466      MOV  #4,$MSGTYPE  ;;TELL APT TO TAKE MSG.
001740 000413      BR   5$
001742 017637 000004 001766      3$: MOV  @4(SP),4$    ;;PUT MSG ADDR IN JSR LINKAGE
001750 062766 000002 000004      ADD  #2,4(SP)      ;;BUMP RETURN ADDRESS
001756 013746 177776      MOV  177776,-(SP) ;;PUSH 177776 ON STACK
001762 004737 044166      JSR  PC,$TYPE     ;;CALL TYPE MACRO
001766 000000      4$: .WORD 0
001770      5$:
001770 105737 002056      10$: TSTB $FFLG       ;;SHOULD REPORT FATAL ERROR?
001774 001416      BEQ  12$          ;;IF NOT: BR
001776 005737 001506      TST  $ENV         ;;RUNNING UNDER APT?
002002 001413      BEQ  12$          ;;IF NOT: BR
002004 005737 001466      11$: TST  $MSGTYPE    ;;FINISHED LAST MESSAGE?
002010 001375      BNE  11$         ;;IF NOT: WAIT
002012 017637 000004 001470      MOV  @4(SP),$FATAL ;;GET ERROR #
002020 062766 000002 000004      ADD  #2,4(SP)      ;;BUMP RETURN ADDR.
002026 005237 001466      INC  $MSGTYPE     ;;TELL APT TO TAKE ERROR
002032 105037 002056      12$: CLRB $FFLG     ;;CLEAR FATAL FLAG
002036 105037 002055      CLRB $LFLG       ;;CLEAR LOG FLAG
002042 105037 002054      CLRB $MFLG       ;;CLEAR MESSAGE FLAG
002046 012601      MOV  (SP)+,R1     ;;POP STACK INTO R1
002050 012600      MOV  (SP)+,R0     ;;POP STACK INTO R0
002052 000207      RTS  PC          ;;RETURN
002054      000      $MFLG: .BYTE 0   ;;MESSG. FLAG
002055      000      $LFLG: .BYTE 0 ;;LOG FLAG
002056      000      $FFLG: .BYTE 0   ;;FATAL FLAG
      .EVEN
      APTSIZE=200
      APTENV=001
      APTPOOL=100
      APTCSUP=040
000200
000001
000100
000040

```

```

575
576
577
578
579 002060 000000
580 002062 000000
581 002064 000000
582 002066 000000
583 002070 000000
584 002072 000000
585
586
587 002074 177570
588 002076 177560
589 002100 177562
590 002102 177564
591 002104 177566
592 002106 000
593 002107 002
594 002110 012
595 002111 000
596 002112 207 377
597 002116 077
598 002117 015
599 002120 012 000
600 002122 377 377 000
601
602
603
604
605 177744
606 177746
607 177750
608 177752
609 177754
610 177776
611 000000
612 000001
613 000002
614 000003
615 000004
616 000005
617 000006
618 000007
619 000001
620 000002
621 000004
622 000010
623 000020
624 000040
625 000100
626 000200
627 000400
628 001000
629 002000
630 004000
631 010000
    
```

```

:*****
: USER LABELS
:*****
    
```

```

$TSTNM: .WORD 0
LOOP: .WORD 0
CMRPAT: .WORD 0
CHRPAT: .WORD 0
FAIL1: .WORD 0
FAIL2: .WORD 0
    
```

```

SWR: .WORD 177570
$TKS: 177560
$TKB: 177562
$TPS: 177564
$TPB: 177566
$NULL: .BYTE 0
$FILLS: .BYTE 2
$FILLC: .BYTE 12
$TPFLG: .BYTE 0
$BELL: .ASCIZ <207><377><377>
$QUES: .ASCII /?/
$CRLF: .ASCII <15>
$LF: .ASCIZ <12>
$ENULL: .BYTE -1,-1,0
    
```

.SBTTL REGISTER DEFINITIONS

```

CME = 177744 ;CACHE MEMORY PARITY FAULT REGISTER
CCR = 177746 ;CACHE CONTROL REGISTER
CMR = 177750 ;CACHE MAINTENANCE REGISTER
CHR = 177752 ;CACHE HIT REGISTER
CDR = 177754 ;CACHE DATA REGISTER
PSW = 177776 ;PROCESSOR STATUS WORD
R0 = %0 ;GENERAL REGISTERS
R1 = %1
R2 = %2
R3 = %3
R4 = %4
R5 = %5
SP = %6
PC = %7
BIT00 = 1
BIT01 = 2
BIT02 = 4
BIT03 = 10
BIT04 = 20
BIT05 = 40
BIT06 = 100
BIT07 = 200
BIT08 = 400
BIT09 = 1000
BIT10 = 2000
BIT11 = 4000
BIT12 = 10000
    
```

632 020000  
633 040000  
634 100000  
635  
636  
637 172300  
638 172302  
639 172304  
640 172306  
641 172310  
642 172312  
643 172314  
644 172316  
645 172340  
646 172342  
647 172344  
648 172346  
649 172350  
650 172352  
651 172354  
652 172356  
653 177572  
654 172516  
655 170200  
656 170202  
657 170204  
658 170206  
659 170210  
660 170212  
661 170214  
662 170216  
663 170220  
664 170222  
665 170002  
666 170004  
667 170000  
668 170006  
669 170016  
670  
671 000200  
672 000001  
673 000100  
674 000040  
675  
676  
677 000001  
678 000004  
679 000010  
680 000100  
681 000200  
682 000400  
683 001000  
684 002000  
685 010000  
686 020000  
687  
688

BIT13 = 20000  
BIT14 = 40000  
BIT15 = 100000  
  
KPDR0 = 172300  
KPDR1 = 172302  
KPDR2 = 172304  
KPDR3 = 172306  
KPDR4 = 172310  
KPDR5 = 172312  
KPDR6 = 172314  
KPDR7 = 172316  
KPAR0 = 172340  
KPAR1 = 172342  
KPAR2 = 172344  
KPAR3 = 172346  
KPAR4 = 172350  
KPAR5 = 172352  
KPAR6 = 172354  
KPAR7 = 172356  
SR0 = 177572  
SR3 = 172516  
UMPR00= 170200  
UMPR01= 170202  
UMPR02= 170204  
UMPR03= 170206  
UMPR04= 170210  
UMPR05= 170212  
UMPR06= 170214  
UMPR07= 170216  
UMPR08= 170220  
UMPR09= 170222  
BECC = 170002  
BEBA = 170004  
BEDA = 170000  
BECR1 = 170006  
BECR2 = 170016

APTSIZE=200  
APTENV=001  
APTSPool=100  
APTCSUP=040

;CCR REGISTER  
DCPI=1  
FMLO=4  
FMHI=10  
WWD=100  
PEA=200  
FC=400  
UCB=1000  
WWPT=2000  
VCIP=10000  
VSIU=20000

;CMR REGISTER

689	000001	TDAR=1
690	000002	HODO=2
691	000004	EHA=4
692	000010	AM=10
693	000020	ESA=20
694	000400	HIT=400
695	001000	TPB=1000
696	002000	LPB=2000
697	004000	HPB=4000
698	010000	VLD=10000
699	020000	CM3=20000
700	040000	CM2=40000
701	100000	CM1=100000
702		
703		:CME REGISTER
704	000040	TPE=40
705	000100	PELO=100
706	000200	PEHI=200
707	100000	CMPE=100000
708		
709	000001	TSTID=1
710	000004	SCPCND=4
711	001015	OFF=1015

```

712
713
714
715
716
717
718
719 002126 000000
720 002130 000000
721 002132 000000
722 002134 000000
723 002136 000000
724 002140 000000
725 002142 000001
726
727
728
729 002144 012737 000340 177776 $SCPSET:MOV #340,PSW ;CPU HI PRIORITY
730 002152 113737 001472 002060 MCVB $TESTN,$STSTM ;MOVE TEST NUMBER TO $STSTM
731 002160 022737 000001 001472 CMP #1,$TESTN ;IS THIS TEST 1?
732 002166 001434 BEQ 3$ ;YES,DO NOT CONSIDER LOOP ON TEST
733 002170 032777 040000 177676 BIT #BIT14,@SWR ;LOOP ON TEST?
734 002176 001413 BEQ 4$ ;NO
735 002200 013702 001472 MOV $TESTN,R2 ;GET PRESENT TEST NUMBER
736 002204 005302 DEC R2 ;GET LAST TEST NUMBER
737 002206 120277 177662 CMPB R2,@SWR ;IS THIS THE TEST?
738 002212 001005 BNE 4$ ;NO
739 002214 005337 001472 5$: DEC $TESTN ;YES;PREPARE FOR LOOP ON TEST
740 002220 013716 002126 MOV STRTST,(SP) ;FUDGE RETURN
741 002224 000002 RTI ;GO LOOP ON TEST
742 002226 005737 001474 4$: TST $PASS ;FIRST PASS?
743 002232 001412 BEQ 3$ ;YES;INHIBIT TEST ITERATIONS
744 002234 032777 004000 177632 BIT #BIT11,@SWR ;INHIBIT ITERATIONS?
745 002242 001006 BNE 3$ ;YES
746 002244 005237 002140 INC TSTCNT ;INCREMENT TEST ITERATION COUNTER
747 002250 023737 002142 002140 CMP TSTIMS,TSTCNT ;ITERATIONS COMPLETE?
748 002256 001356 BNE 5$ ;NO CONTINUE WITH TEST
749 002260 005037 002140 3$: CLR TSTCNT
750 002264 011601 MOV (SP),R1 ;GET ADDRESS OF FIRST ARGUMENT
751 002266 012137 002126 MOV (R1)+,STRTST ;LOCATION OF START OF TEST
752 002272 012137 002130 MOV (R1)+,STRTLP ;LOCATION OF START OF SCOPE LOOP
753 002276 012137 002132 MOV (R1)+,ADRSYNC ;ADDRESS LOADED INTO AMR FOR SCOPE SYNC
754 002302 012137 002134 MOV (R1)+,ADRJMP ;ADDRESS OF END OF SCOPE LOOP AND
755 ;WHERE 'JMP' IS WRITTEN
756 002306 013737 002134 002136 MOV ADRJMP,ADR1$ ;
757 002314 062737 030002 002136 ADD #2,ADR1$ ;LOCATION WHERE '1$' IS WRITTEN
758 002322 012777 000240 177604 MOV #240,@ADRJMP ;INITIALIZE SCOPE LOCATIONS
759 002330 012777 000240 177600 MOV #240,@ADR1$
760 002336 010116 MOV R1,(SP) ;SETUP STACK FOR RETURN
761 002340 000002 RTI

```

```

762
763
764
765
766
767
768 002342 012701 060000
769 002346 012402
770 002350 012421
771 002352 020402
772 002354 001375
773 002356 013721 002366
774 002362 000137 060000
775 002366 000204
776 002370 012701 070000
777 002374 012402
778 002376 012421
779 002400 020402
780 002402 001375
781 002404 013721 002414
782 002410 000137 070000
783 002414 000204
    
```

```

.SBTTL RELOCATION HANDLERS
:*****
: RELOCATION HANDLERS
:*****

RELCTL: MOV #LOW1,R1 ;START OF LOW SPACE
        MOV (R4)+,R2 ;END OF MOVE
1$:     MOV (R4)+,(R1)+ ;TRANSFER TEST
        CMP R4,R2 ;PROCEED TO STOP MARK
        BNE 1$
        MOV 2$, (R1)+ ;RETURN INSTRUCTION
        JMP 60000 ;START TESTS
2$:     RTS R4

RELCTH: MOV #HIGH1,R1 ;START OF HI SPACE
        MOV (R4)+,R2 ;END OF MOVE
1$:     MOV (R4)+,(R1)+ ;TRANSFER TEST
        CMP R4,R2 ;PROCEED TO STOP MARK
        BNE 1$
        MOV 2$, (R1)+ ;RETURN INSTRUCTION
        JMP 70000 ;START TESTS
2$:     RTS R4
    
```

784	002416	012706	000500	BEGIN:	MOV	#500,SP	:SET UP STACK
785	002422	000005			RESET		
786	002424	012737	000340	177776	MOV	#340,PSW	:CPU HI PRIORITY
787	002432	012737	001015	177746	MOV	#OFF,CCR	:DISABLE CACHE
788	002440	005037	001472		CLR	\$TESTN	:RESET TEST ID COUNTER
789	002444	012737	000002	000000	MOV	#2,0	:INITIALIZE A FEW VECTORS
790	002452	005037	000002		CLR	2	
791	002456	012737	000006	000004	MOV	#6,4	
792	002464	005037	000006		CLR	6	
793	002470	012737	000116	000114	MOV	#116,114	
794	002476	005037	000116		CLR	116	
795	002502	005037	002062		CLR	LOOP	:SOFTWARE DELAY
796	002506	005337	002062	1\$:	DEC	LOOP	
797	002512	001375			BNE	1\$	



802

```
.SBTTL TEST # 1 - CACHE REGISTER RESPONSE TESTS
:*****
:*TEST 1 - CACHE REGISTER RESPONSE TESTS
:*   ATTEMPT READ INTO CME TO TEST ADDRESS SELECT LOGIC
:*   IF TIME OUT OCCURES THEN LOGIC IN FAULT
:*****
```

```
TST1:
803 002514 000004      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      002516 002526      .WORD 40$          ;ERROR/LOOP ON TEST
      002520 002526      .WORD 1$          ;TEST START LOCATION
      002522 000000      .WORD 0           ;LOOP ON ERROR START LOCATION
      002524 002552      .WORD 25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      002526              .WORD 25$         ;LOOP ON ERROR END LOCATION
804 002526 012737 002560 000004 40$:      MOV #2$,4          ;SETUP TRAP VECTOR
805 002534 012737 000340 000006 1$:      MOV #340,6
806 002542 005737 177744          TST CME          ;READ PARITY REGISTER
807 002546 000240          NOP
808 002550 000240          NOP
809 002552 000240          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
      002554 000240          NOP          ;FOR LOOP ON ERROR
810 002556 000411          BR 10$          ;NO FAULT;GO TO NEXT TEST
811 002560 022626          CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
812 002562 012737 000006 000004 2$:      MOV #6,4          ;RESTORE TRAP VECTOR
813 002570 005037 000006          CLR 6
814 002574 104406          ERROR          ;ERROR
      002576 002574          .WORD -2          ;-----
815          ;CACHE REGISTER RESPONSE TESTS
816          ;READING PARITY FAULT REGISTER
817          ;CAUSED TIMEOUT
818
819 002600 000000          .WORD 0
820 002602 000240          NOP
      002604 005237 001472 10$:      INC $TESTN      ;END OF TEST
      ;INCREMENT TEST COUNTER
```

825

```
.SBTTL TEST # 2 - READ CCR TO CHECK ADDRESS SELECT LOGIC
:*****
:*TEST 2 - READ CCR TO CHECK ADDRESS SELECT LOGIC
:* ATTEMPT READ INTO CCR TO CHECK ADDRESS SELECT LOGIC
:* IF TIME OUT OCCURES THEN LOGIC IN FAULT
:*****
TST2:
```

```
826 002610 000004          SCLPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          002612 002622          .WORD 40$          ;ERROR/LOOP ON TEST
          002614 002622          .WORD 1$          ;TEST START LOCATION
          002616 000000          .WORD 0          ;LOOP ON ERROR START LOCATION
          002620 002646          .WORD 25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          002622          40$:          .WORD 25$         ;LOOP ON ERROR END LOCATION
827 002622 012737 002654 000004 1$: MOV #2$,4          ;SETUP TRAP VECTOR
828 002630 012737 000340 000006 MOV #340,6
829 002636 005737 177746 TST CCR          ;READ CACHE CONTROL REGISTER
830 002642 000240 NOP
831 002644 000240 NOP
832 002646 000240 25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          002650 000240 NOP          ;FOR LOOP ON ERROR
833 002652 000411 BR 10$          ;NO FAULT;GO TO NEXT TEST
834 002654 022626 2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
835 002656 012737 000006 000004 MOV #6,4          ;RESTORE TRAP VECTOR
836 002664 005037 000006 CLR 6
837 002670 104406 ERROR          ;ERROR
          002672 002670          .WORD -2          ;-----
838          ;CACHE REGISTER RESPONSE TESTS
839          ;READING CACHE CONTROL REGISTER
840          ;CAUSED TIMEOUT
841
842 002674 000000          .WORD 0
843 002676 000240 10$: NOP          ;END OF TEST
          002700 005237 001472 INC $TESTN        ;INCREMENT TEST COUNTER
```

848

```
.SBTTL TEST # 3 - READ CMR TO CHECK ADDRESS SELECT LOGIC
:*****
:*TEST 3 - READ CMR TO CHECK ADDRESS SELECT LOGIC
:*  ATTEMPT READ INTO CMR TO CHECK ADDRESS SELECT LOGIC
:*  IF TIME OCCURES THEN LOGIC IN FAULT
:*****
```

```
TST3:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
849 002704 000004           ;ERROR/LOOP ON TEST
      .WORD 40$             ;TEST START LOCATION
      .WORD 1$              ;LOOP ON ERROR START LOCATION
      .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$             ;LOOP ON ERROR END LOCATION
40$:
850 002716 012737 002750 000004 1$: MOV #2$,4           ;SETUP TRAP VECTOR
851 002724 012737 000340 000006 MOV #340,6
852 002732 005737 177750 TST CMR           ;READ MAINTENANCE REGISTER
853 002736 000240 NOP
854 002740 000240 NOP
855 002742 000240 25$: NOP           ;INSTRUCTION 'JMP 1$' PLACED HERE
      .WORD 1$              ;FOR LOOP ON ERROR
856 002746 000411 BR 10$           ;NO FAULT;GO TO NEXT TEST
857 002750 022626 2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
858 002752 012737 000006 000004 MOV #6,4           ;RESTORE TRAP VECTOR
859 002760 005037 000006 CLR 6
860 002764 104406 ERROR           ;ERROR
      .WORD -2             ;-----
861 002766 002764           ;CACHE REGISTER RESPONSE TESTS
862           ;READING MAINTENANCE REGISTER
863           ;CAUSED TIMEOUT
864 002770 000000 .WORD 0
865 002772 000240 10$: NOP           ;END OF TEST
      .WORD 0                ;INCREMENT TEST COUNTER
      INC $TESTN
```

870

```
.SBTTL TEST # 4 - READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
:*****
:*TEST 4 - READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
:*   ATTEMPT READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
:*   IF TIME OUT OCCURES THEN LOGIC IN FAULT
:*****
```

871 003000 000004

```
TST4:
      SCPCND
      .WORD 40$
      .WORD 1$
      .WORD 0
      .WORD 25$
      :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      :ERROR/LOOP ON TEST
      :TEST START LOCATION
      :LOOP ON ERROR START LOCATION
      :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      :LOOP ON ERROR END LOCATION
```

003002 003012  
 003004 003012  
 003006 000000  
 003010 003036  
 003012

872 003012 012737 003044 000004  
 873 003020 012737 000340 000006  
 874 003026 005737 177752  
 875 003032 000240  
 876 003034 000240  
 877 003036 000240

```
40$:
1$:  MOV #2$,4
     MOV #340,6
     TST CHR
     :SETUP TRAP VECTOR
     NOP
     NOP
     :READ HIT REGISTER
```

003040 000240  
 878 003042 000411  
 879 003044 022626  
 880 003046 012737 000006 000004  
 881 003054 005037 000006  
 882 003060 104406

```
25$:
     NOP
     NOP
     :INSTRUCTION 'JMP 1$' PLACED HERE
     :FOR LOOP ON ERROR
     BR 10$
     :NO FAULT;GO TO NEXT TEST
2$:  CMP (SP)+,(SP)+
     MOV #6,4
     :READJUST STACK DUE TO INTERRUPT
     CLR 6
     :RESTORE TRAP VECTOR
     ERROR
     :ERROR
     :-----
```

003062 003060

883  
 884  
 885

886 003064 000000  
 887 003066 000240  
 003070 005237 001472

```

     .WORD -2
     :CACHE REGISTER RESPONSE TESTS
     :READING HIT REGISTER
     :CAUSED TIMEOUT
10$: .WORD 0
     NOP
     :END OF TEST
     INC $TESTN
     :INCREMENT TEST COUNTER
```

892

```
.SBTTL TEST # 5 - READ CDR TO CHECK ADDRESS SELECT LOGIC
*****
*TEST 5 - READ CDR TO CHECK ADDRESS SELECT LOGIC
* ATTEMPT READ INTO CDR TO CHECK ADDRESS SELECT LOGIC
* IF TIMEOUT OCCURS THEN LOGIC IN FAULT
*****
```

893 003074 000004

TST5:

SCPCND

```
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

003076 003106  
003100 003106  
003102 000000  
003104 003132

```
.WORD 40$  
.WORD 1$  
.WORD 0  
.WORD 25$
```

894 003106 012737 003140 000004  
895 003114 012737 000340 000006  
896 003122 005737 177754

40\$:  
1\$:

```
MOV #2$,4  
MOV #340,6  
TST CDR
```

```
;SETUP TRAP VECTOR  
;READ DATA REGISTER
```

897 003126 000240  
898 003130 000240  
899 003132 000240

25\$:

NOP

```
;INSTRUCTION 'JMP 1$' PLACED HERE  
;FOR LOOP ON ERROR
```

900 003136 000411  
901 003140 022626

2\$:

```
BR 10$  
CMP (SP)+,(SP)+
```

```
;NO FAULT;GO TO NEXT TEST  
;READJUST STACK DUE TO INTERRUPT  
;RESTORE TRAP VECTOR
```

902 003142 012737 000006 000004  
903 003150 005037 000006  
904 003154 104406

```
MOV #6,4  
CLR 6  
ERROR
```

```
;ERROR  
;-----
```

905 003156 003154

```
.WORD -2
```

```
;CACHE REGISTER RESPONSE TESTS  
;READING DATA REGISTER  
;CAUSED TIMEOUT
```

906  
907

908 003160 000000  
909 003162 000240  
003164 005237 001472

10\$:

```
.WORD 0  
NOP  
INC $TESTN
```

```
;END OF TEST  
;INCREMENT TEST COUNTER
```

917

```
.SBTTL TEST # 6 - TEST ADRS SEL LOGIC - WRITE 1 TO BIT 0 OF CME
*****
*TEST 6 - TEST ADRS SEL LOGIC - WRITE 1 TO BIT 0 OF CME
* TESTING ADDRESS SELECTION LOGIC BY WRITING ONE INTO UNUSED
* CME REGISTER BIT00 THEN READ CONTENTS OF REGISTER BACK
* LOOKING TO SEE IF BIT00 STILL READS AS 0.
* IF BIT00 IS SET IT IS POSSIBLE WE ARE ADDRESSING THE WRONG
* REGISTER
*****
```

918 003170 000004

TST6:

SCPCND

```
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

003172 003202  
 003174 003202  
 003176 000000  
 003200 003210

.WORD 40\$  
 .WORD 1\$  
 .WORD 0  
 .WORD 25\$

919 003202 012737 000001 177744

40\$:

MOV #1,CME

```
;WRITE 1 INTO BIT00
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
```

920 003210 000240

1\$:

NOP

003212 000240

NOP

921 003214 032737 000001 177744

25\$:

BIT #1,CME

```
;CHECK FOR 0
```

922 003222 001403

BEQ 10\$

```
;PASS;NEXT TEST
```

923 003224 104406

ERROR

```
;ERROR
```

003226 003224

.WORD -2

```
;-----
```

924

```
;CACHE REGISTER RESPONSE TESTS
```

925

```
;UNUSED CME BIT00 READ AS 1
```

926

```
;POSSIBLE REG. ADDRESS ERROR
```

927 003230 000000

.WORD 0

928 003232 000240

10\$:

NOP

```
;END OF TEST
```

003234 005237 001472

INC \$TESTN

```
;INCREMENT TEST COUNTER
```

935

.SBTTL TEST # 7 - TEST BIT 0 OF DCPI & BIT 1 OF CCR  
\*\*\*\*\*  
\*TEST 7 - TEST BIT 0 OF DCPI & BIT 1 OF CCR  
\* ASSURING BIT00(DCPI) READS AS A 1 AND TESTING ADDRESS  
\* SELECT LOGIC BY WRITING 1 INTO BIT00 OF CCR AND THEN  
\* READING A 1. IF BIT00 READS AS 0 POSSIBLE ADDRESSING  
\* WRONG REGISTER OR CCR REGISTER/DATA PATH ARE BAD.  
\*\*\*\*\*

936 003240 000004

TST7: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION

003242 003252  
003244 003252  
003246 000000  
003250 003260  
003252

937 003252 032737 000001 177746 40\$:  
938 003260 000240 1\$:  
003262 000240 25\$:

BIT #1,CCR ;CHECK BIT00 FOR 1  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
BNE 10\$ ;PASS;NXT TEST  
ERROR ;ERROR  
;-----

003270 003266

.WORD -2 ;CACHE REGISTER RESPONSE TESTS  
;WROTE 1 INTO BIT00 CCR; READ 0

941  
942  
943 003272 000000  
944 003274 000240  
003276 005237 001472

10\$: .WORD 0 ;END OF TEST  
NOP ;INCREMENT TEST COUNTER  
INC \$TESTN

949

```
.SBTTL TEST # 10 - CACHE CONTROL REGISTER DATA TEST (CCR)
*****
*TEST 10 - CACHE CONTROL REGISTER DATA TEST (CCR)
*   VERIFY THAT CCR BIT12(VCIP) READS AS A 0, SINCE A CLEARING
*   OF VALID STORE SHOULD NOT BE HAPPENING AT THIS TIME
*****
```

950 003302 000004

TST10:

SCPCND

```
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

003304 003314

.WORD 40\$

003306 003314

.WORD 1\$

003310 000000

.WORD 0

003312 003322

.WORD 25\$

003314

40\$:

951 003314 032737 010000 177746

1\$:

BIT #BIT12,CCR

```
;CHECK FOR 0
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
```

952 003322 000240

25\$:

NOP

003324 000240

NOP

953 003326 001403

BEQ 10\$

```
;PASS
```

954 003330 104406

ERROR

```
;ERROR
```

003332 003330

.WORD -2

```
;CCR DATA TEST
;READ 1 FROM CCR BIT12. A CLEARING OF
;VALID STORE AT THIS TIME SHOULD NOT
;BE INDICATED
```

955

956

957

958

959 003334 000000

.WORD 0

960 003336 000240

10\$:

NOP

```
;END OF TEST
```

003340 005237 001472

INC \$TESTN

```
;INCREMENT TEST COUNTER
```



966

```
.SBTTL TEST # 11 - TEST BIT 0 OF CCR
*****
:TEST 11 - TEST BIT 0 OF CCR
:* WRITE ZERO INTO CCR BIT00 THEN READ CCR
:* IF CCR IS READ AS ONE THEN CACHE CCR REGISTER MAY BE BAD
:* OR CACHE REGISTER DATA PATH COULD BE IN ERROR
*****
```

```
TST11:
          SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          ;TEST START LOCATION
          ;LOOP ON ERROR START LOCATION
          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          ;LOOP ON ERROR END LOCATION
          40$:
967 003344 000004                .WORD 40$
          003346 003356          .WORD 1$
          003350 003364          .WORD 0
          003352 000000          .WORD 25$
          003354 003376
          003356
968 003356 012737 001415 177746  MOV #OFF+BIT08,CCR ;DISABLE AND FLUSH CACHE
969 003364 042737 000001 177746  BIC #BIT00,CCR ;WRITE 0
970 003372 013700 177746        MOV CCR,RO ;SAVE CCR CONTENTS
971 003376 000240                NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
          003400 000240                NOP ;FOR LOOP ON ERROR
972 003402 032700 000001        BIT #BIT00,RO ;CHECK FOR 0
973 003406 001403                BEQ 10$ ;PASS; NXT TEST
974 003410 104406                ERROR ;ERROR
          ;-----
          003412 003410          .WORD -2 ;CCR DATA TEST
975
976                                ;WROTE 0 INTO BIT00 CCR; READ 1
977 003414 000000                .WORD 0
978 003416 000240                NOP ;END OF TEST
          003420 005237 001472        INC $TESTN ;INCREMENT TEST COUNTER
```

984

.SBTTL TEST # 12 - TEST CLEARING OF BIT 2 OF CCR  
:\*\*\*\*\*  
:TEST 12 - TEST CLEARING OF BIT 2 OF CCR  
:WRITE ZERO INTO CCR BIT02(FMLO) THEN READ CCR  
:IF BIT02 IS READ AS ONE THEN CCR REGISTER MAY BE BAD  
:OR CACHE REGISTER DATA PATH MAY BE AT FAULT  
:\*\*\*\*\*

985 003424 000004

TST12: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 18-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
40\$: MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

003426 003436  
003430 070000  
003432 000000  
003434 070006  
003436 012737  
003444 004437  
003450 003514

001015 177746  
002370

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

986 003452 042737 000004 177746 1\$:  
987 003460 000240 25\$:  
003462 000240  
988 003464 013701 177746  
989 003470 012737 001015 177746  
990 003476 032701 000004  
991 003502 001403  
992 003504 104406

BIC #BIT02,CCR ;WRITE 0  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
MOV CCR,R1 ;SAVE CCR CONTENTS  
MOV #OFF,CCR ;DISABLE CACHE  
BIT #BIT02,R1 ;CHECK FOR 0  
BEQ 10\$ ;PASS; NXT TEST  
ERROR ;ERROR  
:-----

003506 003504

.WORD -2 ;CCR DATA TEST  
;WROTE 0 INTO CCR BIT02; READ 1

993  
994  
995 003510 000000  
996 003512 000240  
003514 005237 001472

10\$: .WORD 0 ;END OF TEST  
NOP ;INCREMENT TEST COUNTER  
INC \$TESTN

1002

```
.SBTTL TEST # 13 - TEST SETTING OF BIT 2 OF CCR
:*****
:*TEST 13 - TEST SETTING OF BIT 2 OF CCR
:* WRITE ONE INTO CCR BIT02(FMLO) AND ASSURE THAT IT READS 1.
:* IF READS BACK AS 0 THEN CCR MAY BE BAD OR CACE DATA PATH
:* IS AT FAULT
:*****
```

1003 003520 000004

```
TST13:
SCPCND
:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
```

003522 003532  
 003524 003532  
 003526 000000  
 003530 003540  
 003532

```
40$:
1$:
25$:
```

1004 003532 052737 000004 177746  
 1005 003540 000240

```
BIS #FMLO,CCR
NOP
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
```

003542 000240  
 1006 003544 032737 000004 177746  
 1007 003552 001003  
 1008 003554 104406

```
BIT #BIT02,CCR
BNE 10$
ERROR
```

```
:CHECK FOR 1
:PASS
:ERROR
:-----
```

003556 003554

```
.WORD .-2
```

1009  
 1010

```
:CCR DATA TEST
:WROTE 1 INTO CCR BIT02; READ 0
```

1011 003560 000000  
 1012 003562 000240  
 003564 005237 001472

```
10$:
NOP
INC $TESTN
```

```
:END OF TEST
:INCREMENT TEST COUNTER
```

1018

```
.SBTTL TEST # 14 - TEST CLEARING OF CCR BIT 3
*****
*TEST 14 - TEST CLEARING OF CCR BIT 3
*   WRITE ZERO INTO CCR BIT03(FMHI) THEN READ CCR
*   IF BIT03 READ BACK AS ONE THEN CCR REGISTER BIT MAY BE BAD
*   OR CACHE REGISTER DATA PATH MAY BE AT FAULT
*****
```

1019 003570 000004

003572 003602  
 003574 060000  
 003576 000000  
 003600 060012  
 003602 012737  
 003610 004437  
 003614 003660

001015 177746  
 002342

```
TST14:
SCPCND
        .WORD 40$
        .WORD 1$-40$+57764
        .WORD 0
        .WORD 25$-40$+57764
MOV     #OFF,CCR
JSR    R4,RELCTL
        .WORD 10$+2
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;DISABLE CACHE
;LOCATE TEST CODE TO LOW CACHE SPACE
;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE
```

1020 003616 042737 000010 177746  
 1021 003624 013700 177746  
 1022 003630 000240  
 003632 000240  
 1023 003634 012737 001015 177746  
 1024 003642 032700 000010  
 1025 003646 001403  
 1026 003650 104406

```
1$:   BIC    #BIT03,CCR
      MOV    CCR,R0
      ;WRITE 0
      ;SAVE CONTENTS OF CCR
25$:  NOP
      NOP
      ;INSTRUCTION 'JMP 1$' PLACED HERE
      ;FOR LOOP ON ERROR
      MOV    #OFF,CCR
      ;DISABLE CACHE
      BIT    #BIT03,R0
      ;CHECK FOR 0
      BEQ   10$
      ;PASS
      ERROR
      ;-----
```

1027 003652 003650

1028

1029 003654 000000  
 1030 003656 000240  
 003660 005237 001472

```
.WORD -.2
;CCR DATA TEST
;WROTE 0 INTO CCR BIT03; READ 1
10$:  .WORD 0
      NOP
      INC   $TESTN
;END OF TEST
;INCREMENT TEST COUNTER
```

1036

```
.SBTTL TEST # 15 - TEST SETTING OF CCR BIT 3
*****
:TEST 15 - TEST SETTING OF CCR BIT 3
:* WRITE 1 INTO CCR BIT03( MHI) AND ASSURE IT READS 1.
:* IF CCR BIT03 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
:* OR CACHE REGISTER DATA PATH MAY BE AT FAULT
*****
```

1037 003664 000004

003666 003676  
 003670 003676  
 003672 000000  
 003674 003704

```
TST15:
SCPCND
:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
```

1038 003676 052737 000010 177746  
 1039 003704 000240

```
40$:
1$: BIS #FMHI,CCR
25$: NOP
```

```
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
```

1040 003710 032737 000010 177746  
 1041 003716 001003  
 1042 003720 104406

```
NOP
BIT #BIT03,CCR
BNE 10$
ERROR
```

```
:CHECK FOR 1
:PASS
:ERROR
:-----
```

003722 003720

```
.WORD .-2
```

```
:CCR DATA TEST
:WROTE 1 INTO CCR BIT03; READ 0
```

1043  
 1044  
 1045 003724 000000  
 1046 003726 000240  
 003730 005237 001472

```
10$: .WORD 0
NOP
INC $TESTN
```

```
:END OF TEST
:INCREMENT TEST COUNTER
```

1052

```
.SBTTL TEST # 16 - TEST CLEARING OF BIT 6 OF CCR
:*****
:*TEST 16 - TEST CLEARING OF BIT 6 OF CCR
:*  WRITE 0 INTO CCR BIT06(WWPD) THEN READ CCR
:*  IF BIT06 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
:*  OR CACHE REGISTER DATA PATH MAY BE AT FAULT
:*****
```

1053 003734 000004

```
TST16: SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
```

003736 003746  
 003740 003746  
 003742 000000  
 003744 003754  
 003746

```
40$:  

1$: BIC #BIT06,CCR  

25$: NOP
```

```
:WRITE 0  

:INSTRUCTION 'JMP 1$' PLACED HERE  

:FOR LOOP ON ERROR  

:CHECK FOR 0  

:PASS  

:ERROR  

:-----
```

1054 003746 042737 000100 177746  
 1055 003754 000240  
 003756 000240  
 1056 003760 032737 000100 177746  
 1057 003766 001403  
 1058 003770 104406

```
BIT #BIT06,CCR  

BEQ 10$  

ERROR
```

003772 003770

```
.WORD -2
```

```
:CCR DATA TEST  

:WROTE 0 INTO CCR BIT06; READ 1
```

1059  
 1060  
 1061 003774 000000  
 1062 003776 000240  
 004000 005237 001472

```
10$: .WORD 0  

NOP  

INC $TESTN
```

```
:END OF TEST  

:INCREMENT TEST COUNTER
```

1068

```
.SBTTL TEST # 17 - TEST CLEARING OF BIT 7 OF CCR
*****
*TEST 17 - TEST CLEARING OF BIT 7 OF CCR
*   WRITE ZERO INTO CCR BIT07(PEA) THEN READ CCR
*   IF CCR BIT07 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
*   OR CACHE REGISTER DATA PATH MAY BE AT FAULT
*****
```

```
1069 004004 000004          SCPCND          :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                :ERROR/LOOP ON TEST
                                :TEST START LOCATION
                                :LOOP ON ERROR START LOCATION
                                :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                :LOOP ON ERROR END LOCATION
                                40$:
                                1$:
                                25$:
1070 004016 042737 000200 177746 40$: BIC      #BIT07,CCR      :WRITE 0
1071 004024 000240          :NOP              :INSTRUCTION 'JMP 1$' PLACED HERE
                                :FOR LOOP ON ERROR
1072 004030 032737 000200 177746      BIT      #BIT07,CCR      :CHECK FOR 0
1073 004036 001403          BEQ       10$          :PASS
1074 004040 104406          ERROR              :ERROR
                                :-----
                                :CCR DATA TEST
                                :WROTE 0 INTO CCR BIT07; READ 1
1075 004042 004040          .WORD     .-2
1076
1077 004044 000000          .WORD     0
1078 004046 000240          10$:  NOP
                                :END OF TEST
                                004050 005237 001472  INC      $TESTN      :INCREMENT TEST COUNTER
```

1084

```
.SBTTL TEST # 20 - TEST SETTING OF BIT 7 OF CCR
*****
*TEST 20 - TEST SETTING OF BIT 7 OF CCR
*  WRITE ONE INTO CCR BIT07 THEN READ CCR
*  IF CCR BIT07 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
*  OR CACHE REGISTER DATA PATH MAY BE AT FAULT
*****
```

```

1085 004054 000004          TST20:      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                           ;ERROR/LOOP ON TEST
                                           ;TEST START LOCATION
004056 004066          .WORD 40$                ;LOOP ON ERROR START LOCATION
004060 004066          .WORD 1$                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004062 000000          .WORD 0                 ;LOOP ON ERROR END LOCATION
004064 004100          .WORD 25$
004066 004066          40$:
1086 004066 052737 000200 177746 1$:  BIS  #BIT07,CCR        ;WRITE 1
1087 004074 013700 177746          MOV  CCR,RO           ;SAVE CCR CONTENTS
1088 004100 000240          25$:  NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
004102 000240          NOP                ;FOR LOOP ON ERROR
1089 004104 012737 001015 177746  MOV  #OFF,CCR        ;DISABLE CACHE
1090 004112 032700 000200          BIT  #BIT07,RO       ;CHECK FOR 1
1091 004116 001003          BNE  10$             ;PASS
1092 004120 104406          ERROR             ;ERROR
                                           ;-----
004122 004120          .WORD  -2
1093                                ;CCR DATA TEST
1094                                ;WROTE 1 INTO CCR BIT07; READ 0
1095 004124 000000          .WORD  0
1096 004126 000240          10$:  NOP                ;END OF TEST
004130 005237 001472          INC  $TESTN         ;INCREMENT TEST COUNTER

```



1102

```
.SBTTL TEST # 21 - TEST CLEARING OF BIT 8 OF CCR
*****
*TEST 21 - TEST CLEARING OF BIT 8 OF CCR
*   WRITE ZERO INTO CCR BIT08(FC) THEN READ CCR
*   IF CCR BIT08 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
*   OR CACHE REGISTER DATA PATH MAY BE AT FAULT
*****
```

```
004134
1103 004134 000004
      004136 004146
      004140 004146
      004142 000000
      004144 004154
      004146
1104 004146 042737 000400 177746
1105 004154 000240
      004156 000240
1106 004160 032737 000400 177746
1107 004166 001401
1108
1109
1110 004170 000000
1111 004172 000240
      004174 005237 001472
```

```
TST21:
SCPCND
      .WORD 40$
      .WORD 1$
      .WORD 0
      .WORD 25$
40$:
1$:
25$:
      BIC #BIT08,CCR
      NOP
      NOP
      BIT #BIT08,CCR
      BEQ 10$
      .WORD 0
10$:
      NOP
      INC $TESTN
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;WRITE 0
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;CHECK FOR 0
;PASS
;CCR DATA TEST
;WROTE 0 INTO CCR BIT08; READ 1
;END OF TEST
;INCREMENT TEST COUNTER
```

1117

```
.SBTTL TEST # 22 - TEST SETTING OF BIT 9 OF CCR
:*****
:*TEST 22 - TEST SETTING OF BIT 9 OF CCR
:*   WRITE 1 INTO CCR BIT09(UCB) AND ASSURE IT READS 1.
:*   IF CCR BIT09 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
:*   OR CACHE REGISTER DATA PATH MAY BE AT FAULT
:*****
```

1118 004200 000004

```
TST22:      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
              ;ERROR/LOOP ON TEST
              ;TEST START LOCATION
              ;LOOP ON ERROR START LOCATION
              ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
              ;LOOP ON ERROR END LOCATION
```

004202 004212  
 004204 004212  
 004206 000000  
 004210 004220  
 004212

1119 004212 052737 001000 177746 40\$:  
 1120 004220 000240 1\$:  
 004222 000240 25\$:

```
      BIS      #UCB,CCR      ;
      NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP      ;FOR LOOP ON ERROR
      BIT      #BIT09,CCR    ;CHECK FOR 1
      BNE     10$           ;PASS
      ERROR    ;ERROR
              ;-----
```

1121 004224 032737 001000 177746  
 1122 004232 C01003  
 1123 004234 104406

004236 004234

1124  
 1125  
 1126 004240 000000  
 1127 004242 000240  
 004244 005237 001472

```
      .WORD   .-2           ;CCR DATA TEST
                              ;WROTE 1 INTO CCR BIT09; READ 0
      .WORD   0             ;
      10$:  NOP            ;END OF TEST
      INC     $TESTN       ;INCREMENT TEST COUNTER
```

1131

```
.SBTTL TEST # 23 - TEST CLEARING BIT 10 OF CCR
:*****
:*TEST 23 - TEST CLEARING BIT 10 OF CCR
:* WRITE ZERO INTO CCR BIT10(WWPT) AND READ 0
:*****
```

```
1132 004250 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                40$:
                                1$: BIC #BIT10,CCR ;WRITE 0
                                25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
                                ;FOR LOOP ON ERROR
                                ;CHECK FOR 0
                                ;PASS
                                ;ERROR
                                ;-----
                                .WORD .-2 ;CCR DATA TEST
                                ;WROTE 0 INTO CCR BIT 10; READ 1
                                10$: .WORD 0 ;END OF TEST
                                NOP ;INCREMENT TEST COUNTER
                                INC $TESTN
```

1146

```
.SBTTL TEST # 24 - CACHE CONTROL REGISTER UNUSED BIT TEST(CCR)
:*****
:*TEST 24 - CACHE CONTROL REGISTER UNUSED BIT TEST(CCR)
:* WRITE INTO UNUSED CCR REGISTER BIT01 THEN READ CCR
:* IF CCR BIT01 READ AS ONE THEN CACHE DATA PATH
:*****
TST24:
```

```
1147 004320 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          004322 004332          .WORD 40$          ;ERROR/LOOP ON TEST
          004324 004332          .WORD 1$          ;TEST START LOCATION
          004326 000000          .WORD 0          ;LOOP ON ERROR START LOCATION
          004330 004340          .WORD 25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          004332          40$:          ;LOOP ON ERROR END LOCATION
1148 004332 052737 000002 177746 1$: BIS #BIT01,CCR ;WRITE 1 INTO UNUSED BIT
1149 004340 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          004342 000240          NOP          ;FOR LOOP ON ERROR
1150 004344 032737 000002 177746 BIT #BIT01,CCR ;CHECK THAT BIT READS 0
1151 004352 001403          BEQ 10$         ;PASS
1152 004354 104406          ERROR          ;ERROR
          004356 004354          .WORD -2         ;-----
1153          ;CCR UNUSED BIT TEST
1154          ;READ 1 FROM UNUSED CCR BIT01
1155          ;SHOULD READ 0
1156 004360 000000          .WORD 0
1157 004362 000240 001472 10$: NOP          ;END OF TEST
          004364 005237          INC $TESTN      ;INCREMENT TEST COUNTER
```

1162

.SBTTL TEST # 25 - TEST UNUSED BIT 4 OF CCR  
\*\*\*\*\*  
\*TEST 25 - TEST UNUSED BIT 4 OF CCR  
\* WRITE ONE INTO UNUSED CCR BIT04 THEN READ CCR  
\* IF CCR BIT04 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR  
\*\*\*\*\*

1163 004370 000004

004372 004402  
004374 004402  
004376 000000  
004400 004410  
004402

1164 004402 052737 000020 177746  
1165 004410 000240  
004412 000240  
1166 004414 032737 000020 177746  
1167 004422 001403  
1168 004424 104406

004426 004424

1169  
1170  
1171

1172 004430 000000  
1173 004432 000240 001472  
004434 005237

TST25: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION  
40\$: .WORD 40\$  
1\$: .WORD 1\$  
25\$: .WORD 0  
;WRITE 1 INTO UNUSED BIT  
;INSTRUCTION 'JMP 1\$' PLACED HERE  
;FOR LOOP ON ERROR  
BIT #BIT04,CCR ;CHECK THAT BIT READS 0  
BEQ 10\$ ;PASS  
ERROR ;ERROR  
;-----  
.WORD -2 ;CCR UNUSED BIT TEST  
;READ 1 FROM UNUSED CCR BIT04  
;SHOULD READ 0  
10\$: .WORD 0  
NOP ;END OF TEST  
INC \$TESTN ;INCREMENT TEST COUNTER

1178

```
.SBTTL TEST # 26 - TEST UNUSED BIT 5 OF CCR
:*****
:*TEST 26 - TEST UNUSED BIT 5 OF CCR
:*   WRITE ONE INTO UNUSED CCR BIT05 THEN READ CCR
:*   IF CCR BIT05 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
:*****
```

1179 004440 000004

```
TST26: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        .WORD 40$ ;ERROR/LOOP ON TEST
        .WORD 1$ ;TEST START LOCATION
        .WORD 0 ;LOOP ON ERROR START LOCATION
        .WORD 25$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        ;LOOP ON ERROR END LOCATION
```

004442 004452  
 004444 004452  
 004446 000000  
 004450 004460  
 004452

1180 004452 052737 000040 177746  
 1181 004460 000240

```
40$: BIS #BIT05,CCR ;WRITE 1 INTO UNUSED BIT
1$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
25$: NOP ;FOR LOOP ON ERROR
```

1182 004464 032737 000040 177746  
 1183 004472 001403  
 1184 004474 104406

```
BIT #BIT05,CCR ;CHECK THAT BIT READS 0
BEQ 10$ ;PASS
ERROR ;ERROR
```

004476 004474

```
.WORD -2
```

1185  
 1186  
 1187

```
;CCR UNUSED BIT TEST
;READ 1 FROM UNUSED CCR BIT05
;SHOULD READ 0
```

1188 004500 000000  
 1189 004502 000240  
 004504 005237 001472

```
10$: .WORD 0 ;END OF TEST
      NOP ;INCREMENT TEST COUNTER
      INC $TESTN
```

1194

```
.SBTTL TEST # 27 - TEST UNUSED BIT 8 OF CCR
:*****
:*TEST 27 - TEST UNUSED BIT 8 OF CCR
:*   WRITE ONE INTO UNUSED CCR BIT08 THEN READ CCR
:*   IF CCR BIT08 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
:*****
TST27:
```

1195	004510	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON :ERROR/LOOP ON TEST :TEST START LOCATION :LOOP ON ERROR START LOCATION :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST :LOOP ON ERROR END LOCATION	
	004512	004522			.WORD	40\$		
	004514	004522			.WORD	1\$		
	004516	000000			.WORD	0		
	004520	004530			.WORD	25\$		
	004522			40\$:				
1196	004522	052737	000400	177746	1\$:	BIS	#BIT08,CCR	:WRITE 1 INTO UNUSED BIT
1197	004530	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	004532	000240				NOP		:FOR LOOP ON ERROR
1198	004534	032737	000400	177746		BIT	#BIT08,CCR	:CHECK THAT BIT READS 0
1199	004542	001403				BEQ	10\$	:PASS
1200	004544	104406				ERROR		:ERROR
	004546	004544				.WORD	.-2	:-----
1201								:CCR UNUSED BIT TEST
1202								:READ 1 FROM UNUSED CCR BIT08
1203								:SHOULD READ 0
1204	004550	000000				.WORD	0	
1205	004552	000240			10\$:	NOP		:END OF TEST
	004554	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER





1226

```
.SBTTL TEST # 31 - TEST UNUSED BIT 14 OF CCR
*****
*TEST 31 - TEST UNUSED BIT 14 OF CCR
*   WRITE ONE INTO UNUSED CCR BIT14 THEN READ CCR
*   IF CCR BIT14 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
*****
```

1227 004630 000004

TST31:

SCPCND

```
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

004632 004642  
 004634 004642  
 004636 000000  
 004640 004650  
 004642

```
.WORD 40$
.WORD 1$
.WORD 0
.WORD 25$
```

1228 004642 052737 040000 177746

40\$:

BIS #BIT14,CCR

```
;WRITE 1 INTO UNUSED BIT
;INSTRUCTION 'JMP 1$' PLACED HERE
```

1229 004650 000240  
 004652 000240

1\$:  
 25\$:

NOP  
 NOP

```
;FOR LOOP ON ERROR
;CHECK THAT BIT READS 0
```

1230 004654 032737 040000 177746

BIT #BIT14,CCR

```
;PASS
;ERROR
```

1231 004662 001403  
 1232 004664 104406

BEQ 10\$  
 ERROR

```
;-----
```

004666 004664

.WORD -2

```
;CCR UNUSED BIT TEST
;READ 1 FROM UNUSED CCR BIT14
;SHOULD READ 0
```

1233  
 1234  
 1235

1236 004670 000000  
 1237 004672 000240

10\$:

```
.WORD 0
NOP
INC $TESTN
```

```
;END OF TEST
;INCREMENT TEST COUNTER
```

004674 005237 001472

1242

```
.SBTTL TEST # 32 - TEST UNUSED BIT 15 OF CCR
*****
*TEST 32 - TEST UNUSED BIT 15 OF CCR
*   WRITE 1 INTO UNUSED CCR BIT15 THEN READ CCR
*   IF CCR BIT15 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
*****
```

1243 004700 000004

```
TST32:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
                          ;TEST START LOCATION
                          ;LOOP ON ERROR START LOCATION
                          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                          ;LOOP ON ERROR END LOCATION
```

004702 004712

```
.WORD 40$
```

004704 004712

```
.WORD 1$
```

004706 000000

```
.WORD 0
```

004710 004720

```
.WORD 25$
```

004712

```
40$:
```

1244 004712 052737 100000 177746

```
1$: BIS #BIT15,CCR
```

```
;WRITE 1 INTO UNUSED BIT
```

1245 004720 000240

```
25$:
```

```
;INSTRUCTION 'JMP 1$' PLACED HERE
```

004722 000240

```
NOP
```

```
;FOR LOOP ON ERROR
```

1246 004724 032737 100000 177746

```
BIT #BIT15,CCR
```

```
;CHECK THAT BIT READS 0
```

1247 004732 001403

```
BEQ 10$
```

```
;PASS
```

1248 004734 104406

```
ERROR
```

```
;ERROR
```

004736 004734

```
.WORD -2
```

```
;CCR UNUSED BIT TEST
```

1249

```
;READ 1 FROM UNUSED CCR BIT15
```

1250

```
;SHOULD READ 0
```

1251

1252 004740 000000

```
.WORD 0
```

1253 004742 000240

```
10$:
```

```
;END OF TEST
```

004744 005237 001472

```
INC $TESTN
```

```
;INCREMENT TEST COUNTER
```

1258

```
.SBTTL TEST # 33 - CME UNUSED BIT 0 TEST
*****
*TEST 33 - CME UNUSED BIT 0 TEST
* ATTEMPT WRITE 1 INTO ALL UNUSED BITS OF CME.
* ALL BITS SHOULD READ 0.
*****
```

1259 004750 000004

004752 004762  
 004754 004762  
 004756 000000  
 004760 004770  
 004762

1260 004762 052737 000001 177744

1261 004770 000240  
 004772 000240

1262 004774 032737 000001 177744

1263 005002 001403  
 1264 005004 104406

1265 005006 005004

1266  
 1267 005010 000000

1268 005012 000240  
 005014 005237 001472

```
TST33:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        .WORD 40$ ;TEST START LOCATION
        .WORD 1$ ;LOOP ON ERROR START LOCATION
        .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 25$ ;LOOP ON ERROR END LOCATION

40$:
1$: BIS #BIT00,CME ;WRITE 1 INTO BIT00
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
    NOP ;FOR LOOP ON ERROR
    BIT #BIT00,CME ;CHECK FOR 0
    BEQ 10$ ;PASS
    ERROR ;ERROR
    -----

        .WORD -2 ;CME UNUSED BIT TEST

        .WORD 0 ;READ 1 FROM UNUSED CME BIT00

10$: NOP ;END OF TEST
    INC $TESTN ;INCREMENT TEST COUNTER
```

1269

```
.SBTTL TEST # 34 - CME UNUSED BIT 1 TEST
:*****
:*TEST 34 - CME UNUSED BIT 1 TEST
:*****
```

```
TST34:
1270 005020 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          005022 005032          .WORD 40$          ;ERROR/LOOP ON TEST
          005024 005032          .WORD 1$          ;TEST START LOCATION
          005026 000000          .WORD 0           ;LOOP ON ERROR START LOCATION
          005030 005040          .WORD 25$        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          005032          40$:          ;LOOP ON ERROR END LOCATION
1271 005032 052737 000002 177744 1$:      BIS #BIT01,CME    ;WRITE 1 INTO BIT01
1272 005040 000240          25$:      NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
          005042 000240          NOP              ;FOR LOOP ON ERROR
1273 005044 032737 000002 177744          BIT #BIT01,CME    ;CHECK FOR 0
1274 005052 001403          BEQ 10$           ;PASS
1275 005054 104406          ERROR              ;ERROR
          005056 005054          .WORD -2         ;-----
1276          ;CME UNUSED BIT TEST
1277          ;READ 1 FROM UNUSED CME BIT01
1278 005060 000000          .WORD 0           ;END OF TEST
1279 005062 000240          10$:      NOP              ;INCREMENT TEST COUNTER
          005064 005237 001472          INC $TESTN
```

1280

```
.SBTTL TEST # 35 - CME UNUSED BIT 2 TEST
:*****
:TEST 35 - CME UNUSED BIT 2 TEST
:*****
```

1281 005070 000004

```
TST35:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        .WORD 40$ ;TEST START LOCATION
        .WORD 1$ ;LOOP ON ERROR START LOCATION
        .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 25$ ;LOOP ON ERROR END LOCATION
```

005072 005102  
 005074 005102  
 005076 000000  
 005100 005110  
 005102

1282 005102 052737 000004 177744  
 1283 005110 000240

```
40$:
1$: BIS #BIT02,CME ;WRITE 1 INTO BIT02
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP ;FOR LOOP ON ERROR
```

1284 005114 032737 000004 177744  
 1285 005122 001403  
 1286 005124 104406

```
BIT #BIT02,CME ;CHECK FOR 0
BEQ 10$ ;PASS
ERROR ;ERROR
:-----
```

005126 005124

```
.WORD -2 ;CME UNUSED BIT TEST
;READ 1 FROM UNUSED CME BIT02
```

1287  
 1288  
 1289 005130 000000  
 1290 005132 000240  
 005134 005237 001472

```
10$: .WORD 0 ;END OF TEST
      NOP ;INCREMENT TEST COUNTER
      INC $TESTN
```

1291

```
.SBTTL TEST # 36 - CME UNUSED BIT 3 TEST
:*****
:TEST 36 - CME UNUSED BIT 3 TEST
:*****
TST36:
```

1292 005140 000004

```
SCPCND                   :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                         :ERROR/LOOP ON TEST
                         :TEST START LOCATION
                         :LOOP ON ERROR START LOCATION
                         :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                         :LOOP ON ERROR END LOCATION
```

005142 005152  
 005144 005152  
 005146 000000  
 005150 005160  
 005152

```
40$:                   BIS     #BIT03,CME
1$:                   NOP
25$:                   NOP
```

```
:WRITE 1 INTO BIT03
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:CHECK FOR 0
:PASS
:ERROR
:-----
```

1293 005152 052737 000010 177744

1294 005160 000240  
 005162 000240

1295 005164 032737 000010 177744

1296 005172 001403  
 1297 005174 104406

```
BIT                   #BIT03,CME
BEQ                   10$
ERROR
```

.WORD .-2

```
:CME UNUSED BIT TEST
:READ 1 FROM UNUSED CME BIT03
```

1298  
 1299

1300 005200 000000  
 1301 005202 000240  
 005204 005237 001472

```
10$:                   NOP
                         INC     $TESTN
```

```
:END OF TEST
:INCREMENT TEST COUNTER
```

1302

```
.SBTTL TEST # 37 - CME UNUSED BIT 4 TEST
:*****
:*TEST 37 - CME UNUSED BIT 4 TEST
:*****
TST37:
```

1303	005210	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON :ERROR/LOOP ON TEST :TEST START LOCATION :LOOP ON ERROR START LOCATION :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST :LOOP ON ERROR END LOCATION	
	005212	005222			.WORD	40\$		
	005214	005222			.WORD	1\$		
	005216	000000			.WORD	0		
	005220	005230			.WORD	25\$		
	005222			40\$:				
1304	005222	052737	000020	177744	1\$:	BIS	#BIT04,CME	:WRITE 1 INTO BIT04
1305	005230	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	005232	000240				NOP		:FOR LOOP ON ERROR
1306	005234	032737	000020	177744		BIT	#BIT04,CME	:CHECK FOR 0
1307	005242	001403				BEQ	10\$	:PASS
1308	005244	104406				ERROR		:ERROR
								:-----
	005246	005244			.WORD	.-2		:CME UNUSED BIT TEST
1309								:READ 1 FROM UNUSED CME BIT04
1310								
1311	005250	000000			.WORD	0		
1312	005252	000240			10\$:	NOP		:END OF TEST
	005254	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

1313

.SBTTL TEST # 40 - CME UNUSED BIT 8 TEST  
 :\*\*\*\*\*  
 :\*TEST 40 - CME UNUSED BIT 8 TEST  
 :\*\*\*\*\*  
 TST40:

1314 005260 000004  
 005262 005272  
 005264 005272  
 005266 000000  
 005270 005300  
 005272  
 1315 005272 052737 000400 177744  
 1316 005300 000240  
 005302 000240  
 1317 005304 032737 000400 177744  
 1318 005312 001403  
 1319 005314 104406  
 005316 005314  
 1320  
 1321  
 1322 005320 000000  
 1323 005322 000240  
 005324 005237 001472

SCPCND  
 .WORD 40\$  
 .WORD 1\$  
 .WORD 0  
 .WORD 25\$  
 40\$:  
 1\$: BIS #BIT08,CME  
 25\$: NOP  
 NOP  
 BIT #BIT08,CME  
 BEQ 10\$  
 ERROR  
 .WORD -2  
 .WORD 0  
 10\$: NOP  
 INC \$TESTN

;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 ;TEST START LOCATION  
 ;LOOP ON ERROR START LOCATION  
 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 ;LOOP ON ERROR END LOCATION  
 ;WRITE 1 INTO BIT08  
 ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 ;FOR LOOP ON ERROR  
 ;CHECK FOR 0  
 ;PASS  
 ;ERROR  
 ;-----  
 ;CME UNUSED BIT TEST  
 ;READ 1 FROM UNUSED CME BIT08  
 ;END OF TEST  
 ;INCREMENT TEST COUNTER



1324

```
.SBTTL TEST # 41 - CME UNUSED BIT 9 TEST
:*****
:*TEST 41 - CME UNUSED BIT 9 TEST
:*****
TST41:
```

1325	005330	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON :ERROR/LOOP ON TEST :TEST START LOCATION :LOOP ON ERROR START LOCATION :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST :LOOP ON ERROR END LOCATION
	005332	005342			.WORD	40\$	
	005334	005342			.WORD	1\$	
	005336	000000			.WORD	0	
	005340	005350			.WORD	25\$	
	005342						
1326	005342	052737	001000	177744	40\$:		
1327	005350	000240			1\$:	BIS	#BIT09,CME
	005352	000240			25\$:	NOP	
1328	005354	032737	001000	177744		NOP	
1329	005362	001403				BIT	#BIT09,CME
1330	005364	104406				BEQ	10\$
						ERROR	
	005366	005364				.WORD	.-2
1331							:CME UNUSED BIT TEST
1332							:READ 1 FROM UNUSED CME BIT09
1333	005370	000000				.WORD	0
1334	005372	000240			10\$:	NOP	
	005374	005237	001472			INC	\$TESTN
							:END OF TEST :INCREMENT TEST COUNTER

1335

```
.SBTTL TEST # 42 - CME UNUSED BIT 10 TEST
:*****
:*TEST 42 - CME UNUSED BIT 10 TEST
:*****
TST42:
```

1336	005400	000004				SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	005402	005412				.WORD 40\$		:ERROR/LOOP ON TEST
	005404	005412				.WORD 1\$		:TEST START LOCATION
	005406	000000				.WORD 0		:LOOP ON ERROR START LOCATION
	005410	005420				.WORD 25\$		:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	005412				40\$:			:LOOP ON ERROR END LOCATION
1337	005412	052737	002000	177744	1\$:	BIS #BIT10,CME		:WRITE 1 INTO BIT10
1338	005420	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	005422	000240				NOP		:FOR LOOP ON ERROR
1339	005424	032737	002000	177744		BIT #BIT10,CME		:CHECK FOR 0
1340	005432	001403				BEQ 10\$		:PASS
1341	005434	104406				ERROR		:ERROR
								:-----
	005436	005434				.WORD -2		:CME UNUSED BIT TEST
1342								:READ 1 FROM UNUSED CME BIT10
1343								
1344	005440	000000				.WORD 0		
1345	005442	000240			10\$:	NOP		:END OF TEST
	005444	005237	001472			INC \$TESTN		:INCREMENT TEST COUNTER

1346

```
.SBTTL TEST # 43 - CME UNUSED BIT 11 TEST
:*****
:*TEST 43 - CME UNUSED BIT 11 TEST
:*****
TST43:
```

1347 005450 000004

SCPCND

```
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

005452 005462  
 005454 005462  
 005456 000000  
 005460 005470  
 005462

```
.WORD 40$
.WORD 1$
.WORD 0
.WORD 25$
```

1348 005462 052737 004000 177744

40\$:

BIS #BIT11,CME

```
;WRITE 1 INTO BIT11
```

1349 005470 000240

1\$:

NOP

```
;INSTRUCTION 'JMP 1$' PLACED HERE
```

005472 000240

NOP

```
;FOR LOOP ON ERROR
```

1350 005474 032737 004000 177744

BIT #BIT11,CME

```
;CHECK FOR 0
```

1351 005502 001403

BEQ 10\$

```
;PASS
```

1352 005504 104406

ERROR

```
;ERROR
```

005506 005504

.WORD -2

```
;CME UNUSED BIT TEST
;READ 1 FROM UNUSED CME BIT11
```

1353

1354

1355 005510 000000

.WORD 0

1356 005512 000240

10\$:

NOP

```
;END OF TEST
```

005514 005237 001472

INC \$TESTN

```
;INCREMENT TEST COUNTER
```

1357

```
.SBTTL TEST # 44 - CME UNUSED BIT 12 TEST
:*****
:*TEST 44 - CME UNUSED BIT 12 TEST
:*****
TST44:
```

1358	005520	000004				SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	005522	005532				.WORD	40\$	:ERROR/LOOP ON TEST
	005524	005532				.WORD	1\$	:TEST START LOCATION
	005526	000000				.WORD	0	:LOOP ON ERROR START LOCATION
	005530	005540				.WORD	25\$	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	005532				40\$:			:LOOP ON ERROR END LOCATION
1359	005532	052737	010000	177744	1\$:	BIS	#BIT12,CME	:WRITE 1 INTO BIT12
1360	005540	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	005542	000240				NOP		:FOR LOOP ON ERROR
1361	005544	032737	010000	177744		BIT	#BIT12,CME	:CHECK FOR 0
1362	005552	001403				BEQ	10\$	:PASS
1363	005554	104406				ERROR		:ERROR
								:-----
	005556	005554				.WORD	.-2	:CME UNUSED BIT TEST
1364								:READ 1 FROM UNUSED CME BIT12
1365								
1366	005560	000000				.WORD	0	
1367	005562	000240			10\$:	NOP		:END OF TEST
	005564	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

1368

```
.SBTTL TEST # 45 - CME UNUSED BIT 13 TEST
:*****
:*TEST 45 - CME UNUSED BIT 13 TEST
:*****
TST45:
```

```
1369 005570 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          005572 005602          .WORD 40$          ;ERROR/LOOP ON TEST
          005574 005602          .WORD 1$          ;TEST START LOCATION
          005576 000000          .WORD 0           ;LOOP ON ERROR START LOCATION
          005600 005610          .WORD 25$        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          005602          40$:          ;LOOP ON ERROR END LOCATION
1370 005602 052737 020000 177744 1$:          BIS #BIT13,CME      ;WRITE 1 INTO BIT13
1371 005610 000240          25$:          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          005612 000240          NOP          ;FOR LOOP ON ERROR
1372 005614 032737 020000 177744          BIT #BIT13,CME    ;CHECK FOR 0
1373 005622 001403          BEQ 10$         ;PASS
1374 005624 104406          ERROR          ;ERROR
          005626 005624          .WORD -2         ;-----
1375          ;CME UNUSED BIT TEST
1376          ;READ 1 FROM UNUSED CME BIT13
1377 005630 000000          .WORD 0
1378 005632 000240          10$:          NOP          ;END OF TEST
          005634 005237 001472          INC $TESTN      ;INCREMENT TEST COUNTER
```

1379

```
.SBTTL TEST # 46 - CME UNUSED BIT 14 TEST
:*****
:*TEST 46 - CME UNUSED BIT 14 TEST
:*****
TST46:
```

1380	005640	000004				SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	005642	005652				.WORD	40\$	:ERROR/LOOP ON TEST
	005644	005652				.WORD	1\$	:TEST START LOCATION
	005646	000000				.WORD	0	:LOOP ON ERROR START LOCATION
	005650	005660				.WORD	25\$	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	005652				40\$:			:LOOP ON ERROR END LOCATION
1381	005652	052737	040000	177744	1\$:	BIS	#BIT14,CME	:WRITE 1 INTO BIT14
1382	005660	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	005662	000240				NOP		:FOR LOOP ON ERROR
1383	005664	032737	040000	177744		BIT	#BIT14,CME	:CHECK FOR 0
1384	005672	001403				BEQ	10\$	:PASS
1385	005674	104406				ERROR		:ERROR
								:-----
	005676	005674				.WORD	.-2	
1386								:CME UNUSED BIT TEST
1387								:READ 1 FROM UNUSED CME BIT14
1388	005700	000000				.WORD	0	
1389	005702	000240			10\$:	NOP		:END OF TEST
	005704	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

1395

```
.SBTTL TEST # 47 - CACHE CONTROL REGISTER BYTE TESTS (CCR)
:*****
:*TEST 47 - CACHE CONTROL REGISTER BYTE TESTS (CCR)
:* REGISTER BYTE SELECTION LOGIC TEST
:* WRITE ONE INTO LOW BYTE WRITE ZERO INTO HIGH BYTE
:* VERIFY THAT LOW BYTE DATA IS NOT EFFECTED BY WRITE TO HIGH BYTE
:*****
```

1396 005710 000004

```
TST47: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

005712 005722  
 005714 005722  
 005716 000000  
 005720 005736  
 005722

```
40$:
1$: BISB #BIT02,CCR ;WRITE 1 INTO CONTROL REGISTER BIT02
BICB #BIT02,CCR+1 ;WRITE 0 INTO CONTROL REGISTER BIT10
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIT #BIT02,CCR ;CHECK FOR 1
BNE 10$ ;PASS
ERROR ;ERROR
;-----
```

1397 005722 152737 000004 177746  
 1398 005730 142737 000004 177747  
 1399 005736 000240  
 005740 000240  
 1400 005742 032737 000004 177746  
 1401 005750 001003  
 1402 005752 104406

005754 005752

```
.WORD -2 ;CACHE CONTROL REGISTER BYTE TESTS
;WROTE ONE INTO LOW BYTE BIT02
;WROTE ZERO INTO HIGH BYTE BIT10
;READ ZERO FROM BIT02
```

1403  
 1404  
 1405  
 1406

1407 005756 000000  
 1408 005760 000240  
 005762 005237 001472

```
10$: .WORD 0
NOP ;END OF TEST
INC $TESTN ;INCREMENT TEST COUNTER
```





1430

```
.SBTTL TEST # 51 - CACHE MAINTENANCE REGISTER DATA TEST (CMR)
*****
:TEST 51 - CACHE MAINTENANCE REGISTER DATA TEST (CMR)
:  VERIFY CMR BIT00(TDAR) CAN BE WRITTEN TO A 0
*****
TST51:
```

1431 006044 000004

SCPCND

```
:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
```

006046 006056  
006050 006056  
006052 000000  
006054 006070

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

1432 006056 042737 000001 177750

40\$:  
1\$:

BIC #BIT00,CMR  
MOV CMR,RO

```
:WRITE 0 INTO CMR BIT00
:SAVE CONTENTS OF CMR
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
```

1433 006064 013700 177750  
1434 006070 000240

25\$:

NOP  
NOP

1435 006074 005037 177750  
1436 006100 032700 000001

CLR CMR  
BIT #BIT00,RO  
BEQ 10\$

```
:CLR MAINT
:CHECK FOR 0 IN BIT00
:PASS
:ERROR
:-----
```

1437 006104 001403  
1438 006106 104406

ERROR

1439 006110 006106

.WORD -2

```
:MAINTENANCE REGISTER DATA TEST
:WROTE 0 INTO CMR BIT00; READ 1
```

1440

1441 006112 000000

.WORD 0

1442 006114 000240 001472  
006116 005237

10\$:

NOP  
INC \$TESTN

```
:END OF TEST
:INCREMENT TEST COUNTER
```

1446

```
.SBTTL TEST # 52 - TEST CMR BIT 0
:*****
:*TEST 52 - TEST CMR BIT 0
:*  VERIFY CMR BIT00(TDAR) CAN BE WRITTEN TO A 1
:*****
```

1447 006122 000004

```
TST52: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
```

006124 006134  
 006126 006134  
 006130 000000  
 006132 006146  
 006134

1448 006134 052737 000001 177750  
 1449 006142 013700 177750  
 1450 006146 000240

```
40$:
1$: BIS #BIT00,CMR ;WRITE 1 INTO CMR BIT00
MOV CMR,RO ;SAVE CONTENTS OF CMR
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLR CMR ;CLR MAINT
BIT #BIT00,RO ;CHECK FOR 1 IN BIT00
BNE 10$ ;PASS
ERROR ;ERROR
;-----
```

006150 000240

1451 006152 005037 177750  
 1452 006156 032700 000001  
 1453 006162 001003  
 1454 006164 104406

006166 006164

1455  
 1456

1457 006170 000000  
 1458 006172 000240  
 006174 005237 001472

```
.WORD .-2 ;MAINTENANCE REGISTER DATA TEST
;WROTE 1 INTO CMR BIT00; READ 0
10$: .WORD 0 ;END OF TEST
NOP ;INCREMENT TEST COUNTER
INC $TESTN
```

1462

.SBTTL TEST # 53 - TEST BIT 1 OF CMR  
 \*\*\*\*\*  
 \*TEST 53 - TEST BIT 1 OF CMR  
 \* VERIFY CMR BIT01(HODO) CAN BE WRITTEN AS A 0.  
 \*\*\*\*\*  
 TST53:

1463 006200 000004  
 006202 006212  
 006204 006212  
 006206 000000  
 006210 006224  
 006212  
 1464 006212 042737 000002 177750  
 1465 006220 013700 177750  
 1466 006224 000240  
 006226 000240  
 1467 006230 005037 177750  
 1468 006234 032700 000002  
 1469 006240 001403  
 1470 006242 104406  
 006244 006242  
 1471  
 1472  
 1473 006246 000000  
 1474 006250 000240  
 006252 005237 001472

SCPCND  
 .WORD 40\$  
 .WORD 1\$  
 .WORD 0  
 .WORD 25\$  
 40\$:  
 1\$: BIC #BIT01,CMR  
 MOV CMR,RO  
 25\$: NOP  
 NOP  
 CLR CMR  
 BIT #BIT01,RO  
 BEQ 10\$  
 ERROR  
 .WORD .-2  
 10\$: .WORD 0  
 NOP  
 INC \$TESTN

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 :ERROR/LOOP ON TEST  
 :TEST START LOCATION  
 :LOOP ON ERROR START LOCATION  
 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 :LOOP ON ERROR END LOCATION  
 :WRITE 0 INTO CMR BIT01  
 :SAVE CONTENTS OF CMR  
 :INSTRUCTION 'JMP 1\$' PLACED HERE  
 :FOR LOOP ON ERROR  
 :CLR MAINT  
 :CHECK FOR 0 IN BIT01  
 :PASS  
 :ERROR  
 :-----  
 :MAINTENANCE REGISTER DATA TEST  
 :WROTE 0 INTO CMR BIT01; READ 1  
 :END OF TEST  
 :INCREMENT TEST COUNTER

1478

```
.SBTTL TEST # 54 - TEST CMR BIT 1  
*****  
*TEST 54 - TEST CMR BIT 1  
*   VERIFY CMR BIT01(MODO) CAN BE WRITTEN AS A 1.  
*****
```

```
1479 006256 000004  
      006260 006270  
      006262 006270  
      006264 000000  
      006266 006302  
      006270  
1480 006270 052737 000002 177750  
1481 006276 013700 177750  
1482 006302 000240  
      006304 000240  
1483 006306 005037 177750  
1484 006312 032700 000002  
1485 006316 001003  
1486 006320 104406  
  
      006322 006320  
1487  
1488  
1489 006324 000000  
1490 006326 000240  
      006330 005237 001472
```

```
TST54:  
      SCPCND  
      .WORD 40$  
      .WORD 1$  
      .WORD 0  
      .WORD 25$  
40$:  
1$:  BIS #BIT01,CMR  
      MOV CMR,RO  
25$:  NOP  
      NOP  
      CLR CMR  
      BIT #BIT01,RO  
      BNE 10$  
      ERROR  
  
      .WORD -2  
  
10$: .WORD 0  
      NOP  
      INC $TESTN  
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION  
;WRITE 1 INTO CMR BIT01  
;SAVE CONTENTS OF CMR  
;INSTRUCTION 'JMP 1$' PLACED HERE  
;FOR LOOP ON ERROR  
;CLR MAINT  
;CHECK FOR 1 IN BIT01  
;PASS  
;ERROR  
;-----  
;MAINTENANCE REGISTER DATA TEST  
;WROTE 1 INTO CMR BIT01; READ 0  
;END OF TEST  
;INCREMENT TEST COUNTER
```

1494

.SBTTL TEST # 55 - TEST CMR BIT 3

\*\*\*\*\*  
\*TEST 55 - TEST CMR BIT 3  
\* VERIFY CMR BIT03(AM) CAN BE WRITTEN AS A 0.  
\*\*\*\*\*

1495 006334 000004

TST55:

SCPCND

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
:ERROR/LOOP ON TEST  
:TEST START LOCATION  
:LOOP ON ERROR START LOCATION  
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
:LOOP ON ERROR END LOCATION

006336 006346  
006340 006346  
006342 000000  
006344 006372

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

1496 006346 012737 177777 177752  
1497 006354 112737 000374 177751  
1498 006362 105037 177750  
1499 006366 013700 177750

40\$:  
1\$:

MOV #-1,CHR  
MOVB #374,CMR+1  
CLRB CMR  
MOV CMR,RO

:ALL 1'S TO AMR

1500 006372 000240  
006374 000240  
1501 006376 005037 177750  
1502 006402 032700 000010  
1503 006406 001403  
1504 006410 104406

25\$:

NOP  
NOP  
CLR CMR  
BIT #BIT03,RO  
BEQ 10\$  
ERROR

:WRITE 0 INTO CMR BIT03  
:SAVE CONTENTS OF CMR  
:INSTRUCTION 'JMP 1\$' PLACED HERE  
:FOR LOOP ON ERROR  
:CLR MAINT  
:CHECK FOR 0 IN BIT03  
:PASS  
:ERROR  
:-----

006412 006410

.WORD -2

1505  
1506  
1507 006414 000000  
1508 006416 000240 001472  
006420 005237

10\$:

.WORD 0  
NOP  
INC \$TESTN

:MAINTENANCE REGISTER DATA TEST  
:WROTE 0 INTO CMR BIT03; READ 1  
:END OF TEST  
:INCREMENT TEST COUNTER

1512

```
.SBTTL TEST # 56 - TEST CMR BIT 3  
:*****  
:*TEST 56 - TEST CMR BIT 3  
:*   VERIFY BIT03(AM) CAN BE WRITTEN AS A 1.  
:*****
```

1513 006424 000004

006426 006436  
006430 006436  
006432 000000  
006434 006470  
006436

1514 006436 012737 177777 177752  
1515 006444 112737 000374 177751  
1516 006452 105037 177750  
1517 006456 112737 000010 177750  
1518 006464 013700 177750  
1519 006470 000240  
006472 000240  
1520 006474 005037 177750  
1521 006500 032700 000010  
1522 006504 001003  
1523 006506 104406

```
TST56: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION
```

```
40$:  
1$: MOV #-1,CHR  
MOV #374,CMR+1  
CLRB CMR  
MOVB #AM,CMR  
MOV CMR,RO
```

```
25$: NOP ;PRECONDITION AM BIT TO 0  
NOP ;WRITE 1 INTO AM BIT  
CLR CMR ;SAVE CONTENTS OF CMR  
BIT #BIT03,RO ;INSTRUCTION 'JMP 1$' PLACED HERE  
BNE 10$ ;FOR LOOP ON ERROR  
ERROR ;CLR MAINT  
;CHECK FOR 1 IN BIT03  
;PASS  
;ERROR  
;-----
```

006510 006506

1524  
1525  
1526 006512 000000  
1527 006514 000240  
006516 005237 001472

```
.WORD -2 ;MAINTENANCE REGISTER DATA TEST  
;WROTE 1 INTO CMR BIT03; READ 0  
10$: .WORD 0  
NOP ;END OF TEST  
INC $TESTN ;INCREMENT TEST COUNTER
```

1532

.SBTTL TEST # 57 - TEST UNUSED BIT 5 IN THE CMR  
:\*\*\*\*\*  
:TEST 57 - TEST UNUSED BIT 5 IN THE CMR  
:ATTEMPT WRITE 1 INTO ALL UNUSED BITS OF CMR. ALL  
:BITS SHOULD READ 0.  
:\*\*\*\*\*  
TST57:

1533 006522 000004

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION

006524 006534  
006526 006534  
006530 000000  
006532 006542  
006534

40\$:  
1\$: BIS #BIT05,CMR  
25\$: NOP

;WRITE 1 INTO BIT05  
;INSTRUCTION 'JMP 1\$' PLACED HERE  
;FOR LOOP ON ERROR

1534 006534 052737 000040 177750  
1535 006542 000240

006544 000240

BIT #BIT05,CMR  
BEQ 10\$  
ERROR

;CHECK FOR 0  
;PASS  
;ERROR  
;-----

1536 006546 032737 000040 177750  
1537 006554 001403  
1538 006556 104406

006560 006556

.WORD .-2

;CMR UNUSED BIT TEST  
;READ 1 FROM UNUSED CMR BIT05

1539  
1540

006562 000000

10\$: .WORD 0

;END OF TEST  
;INCREMENT TEST COUNTER

1541 006564 000240  
1542 006566 005237 001472

NOP  
INC \$TESTN

1543

.SBTTL TEST # 60 - TEST UNUSED BIT 6 IN THE CMR  
:\*\*\*\*\*  
:\*TEST 60 - TEST UNUSED BIT 6 IN THE CMR  
:\*\*\*\*\*

TST60:

1544	006572	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON :ERROR/LOOP ON TEST :TEST START LOCATION :LOOP ON ERROR START LOCATION :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST :LOOP ON ERROR END LOCATION
	006574	006604			.WORD	40\$	
	006576	006604			.WORD	1\$	
	006600	000000			.WORD	0	
	006602	006612			.WORD	25\$	
	006604						
1545	006604	052737	000100	177750	40\$:		
1546	006612	000240			1\$:	BIS	#BIT06,CMR
	006614	000240			25\$:	NOP	
1547	006616	032737	000100	177750		NOP	
1548	006624	001403				BIT	#BIT06,CMR
1549	006626	104406				BEQ	10\$
						ERROR	
	006630	006626				.WORD	.-2
1550							:CMR UNUSED BIT TEST
1551							:READ 1 FROM UNUSED CMR BIT02
1552	006632	000000				.WORD	0
1553	006634	000240			10\$:	NOP	:END OF TEST
	006636	005237	001472			INC	:INCREMENT TEST COUNTER





.SBTTL TEST # 62 - TEST AMR

```

*****
*TEST 62 - TEST AMR
* MA<21:0> ADDRESS LINES ALL 1'S
* CA<21:0> ADDRESS LINES ALL 0'S
* AMR<21:0> DATA LINES ALL 0'S
* AM BIT SHOULD READ 1.
*****
  
```

```

1572 006712 000004
      006714 006724
      006716 006724
      006720 000000
      006722 006764
      006724
1573 006724 012737 177777 177752
1574 006732 112737 000374 177751
1575 006740 105037 177750
1576 006744 005037 177752
1577 006750 105037 177751
1578 006754 005737 000000
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590 006760 013703 177750
1591 006764 000240
      006766 000240
1592 006770 032703 000010
1593 006774 001003
1594 006776 104406
      007000 006776
1595
1596
1597
1598
1599 007002 000000
1600 007004 000240
      007006 005237 001472
1601
1602
1603
1604
1605
1606 007012 012737 077406 172300
1607 007020 012737 077406 172302
1608 007026 012737 077406 172304
1609 007034 012737 077406 172306
  
```

```

TST62:
SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                ;ERROR/LOOP ON TEST
                ;TEST START LOCATION
                ;LOOP ON ERROR START LOCATION
                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                ;LOOP ON ERROR END LOCATION
40$:
1$:
MOV             #-1,CHR          ;ALL 1'S TO AMR
MOVB           #374,CMR+1
CLRB           CMR
CLR            CHR
CLRB           CMR+1
TST            0
                ;PRECONDITION AM BIT TO 0
                ;ALL 0'S TO AMR<21:0>
                ;PLACE ALL 0'S ON CA<21:0>.HOWEVER,
                ;THIS IS NOT WHEN THE AM BIT IS SET:
                ;WHEN PAX ADDRESS LINES ARE NOT BEING
                ;ACCESSED BY THE CPU, THE CACHE DEFAULTS
                ;TO SELECTING MA<21:0> ADDRESS LINES.
                ;IN THIS SITUATION, MA<21:0> DEFAULTS
                ;TO ALL 1'S THEREBY PLACING ALL 0'S
                ;ON CA<21:0>. THEREFORE, FOLLOWING THE LOADING
                ;OF ALL 0'S INTO AMR<21:0>,AND BEFORE THE
                ;'TST 0' INSTRUCTION, THE AM BIT SHOULD
                ;BE SET DUE TO MATCH BETWEEN AMR<21:0> AND CA<21:0>
                ; ADDRESS LINES.
                ;SAVE AM BIT RESULT IN CMR
                ;INSTRUCTION 'JMP 1$' PLACED HERE
                ;FOR LOOP ON ERROR
                ;AM BIT SHOULD READ 1 INDICATING MATCH
                ;PASS
                ;ERROR
                ;-----
                .WORD          .-2
                ;AMR TESTS
                ;AMR BIT DID NOT READ 1 INDICATING
                ;A MATCH OF ALL 0'S BETWEEN MA TO CA<21:0>
                ;ADDRESS LINES AND AMR<21:0> DATA
                ;ERROR PRINT TERMIN.
                ;END OF TEST
                ;INCREMENT TEST COUNTER
                .SBTTL        MEMORY MANAGEMENT AND UNIBUS MAP REGISTERS SETUP
*****
*          MEMORY MANAGEMENT SETUP
*****
MAGPRE: MOV     #77406,KPDR0    ;ALLOW ALL ACCESS TO KERNEL PAGE 0
          MOV     #77406,KPDR1  ;ALLOW ALL ACCESS TO KERNEL PAGE 1
          MOV     #77406,KPDR2  ;ALLOW ALL ACCESS TO KERNEL PAGE 2
          MOV     #77406,KPDR3  ;ALLOW ALL ACCESS TO KERNEL PAGE 3
  
```

```

1610 007042 012737 077406 172310      MOV      #77406,KPDR4      ;ALLOW ALL ACCESS TO KERNEL PAGE 4
1611 007050 012737 077406 172312      MOV      #77406,KPDR5      ;ALLOW ALL ACCESS TO KERNEL PAGE 5
1612 007056 012737 077406 172314      MOV      #77406,KPDR6      ;ALLOW ALL ACCESS TO KERNEL PAGE 6
1613 007064 012737 077406 172316      MOV      #77406,KPDR7      ;ALLOW ALL ACCESS TO KERNEL PAGE 7
1614 007072 005037 172340      CLR      KPAR0             ;MAP PAGE 0 FOR 0-4K
1615 007076 012737 000200 172342      MOV      #200,KPAR1        ;MAP PAGE 1 FOR 4-8K
1616 007104 012737 000400 172344      MOV      #400,KPAR2        ;MAP PAGE 2 FOR 8-12K
1617 007112 012737 000600 172346      MOV      #600,KPAR3        ;MAP PAGE 3 FOR 12-16K
1618 007120 012737 177600 172356      MOV      #177600,KPAR7     ;MAP PAGE 7 FOR 124-128K
1619
1620
1621
1622
1623 007126 012737 000000 170200      MOV      #0,UMPRO0         ;MAP REGISTER SET 0 FOR 0-4K
1624 007134 012737 000000 170202      MOV      #0,UMPRO1
1625 007142 012737 020000 170204      MOV      #20000,UMPRO2     ;MAP REGISTER SET 1 FOR 4K-8K
1626 007150 012737 000000 170206      MOV      #0,UMPRO3
1627 007156 012737 040000 170210      MOV      #40000,UMPRO4     ;MAP REGISTER SET 2 FOR 8K-12K
1628 007164 012737 000000 170212      MOV      #0,UMPRO5
1629 007172 012737 060000 170214      MOV      #60000,UMPRO6     ;MAP REGISTER SET 3 FOR 12K-16K
1630 007200 012737 000000 170216      MOV      #0,UMPRO7
    
```

```

:*****
:*      UNIBUS MAP REGISTERS SETUP
:*****
    
```

1637

.SBTTL TEST # 63 - AMR CHECK

```

*****
*TEST 63 - AMR CHECK
*  MA<21:0> ADDRESS LINES ALL 0'S
*  CA<21:0> ADDRESS LINES ALL 1'S
*  AMR<21:0> DATA LINES ALL 1'S
*  AM BIT SHOULD READ 1.
*****

```

TST63:

1638 007206 000004

SCPCND

```

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION

```

007210 007220  
007212 007220  
007214 000000  
007216 007244

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

1639 007220 012737 177777 177752  
1640 007226 112737 000374 177751  
1641 007234 105037 177750  
1642 007240 105737 177777

40\$:

1\$:

MOV #-1,CHR  
MOVB #374,CMR+1  
CLRB CMR  
TSTB 177777

```

:LOAD AMR<15:0> WITH 1'S FROM CHR<15:0>
:LOAD AMR<21:16> ALL 1'S
:PRECONDITION AM BIT TO 0
:PUT ALL 1'S ON MA AND PA ADDRESS LINES
:MA WILL BE SELECTED

```

1643 007244 000240

25\$:

NOP  
NOP  
BIT #AM,CMR  
BNE 10\$  
ERROR

```

:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:AM BIT SHOULD READ 1 INDICATING MATCH
:PASS
:ERROR
:-----

```

1644 007246 000240  
1645 007250 032737 000010 177750  
1646 007256 001003  
1647 007260 104406

007262 007260

.WORD -2

1648  
1649  
1650  
1651

```

:AMR TESTS
:AMR BIT DID NOT READ 1 INDICATING
:A MATCH OF ALL 1'S BETWEEN CA<21:0>
:ADDRESS LINES AND AMR<21:0> DATA
:ERROR PRINT TERMIN.

```

1652 007264 000000  
1653 007266 000240  
007270 005237 001472

10\$:

.WORD 0  
NOP  
INC \$TESTN

```

:END OF TEST
:INCREMENT TEST COUNTER

```

TEST # 64 - AMR LINES NOT SHORTED & NOT SHORTED TO CA LINES  
1660

.SBTTL TEST # 64 - AMR LINES NOT SHORTED & NOT SHORTED TO CA LINES  
\*\*\*\*\*  
\*TEST 64 - AMR LINES NOT SHORTED & NOT SHORTED TO CA LINES  
\* MA<21:0> ADDRESS LINES ALL 1'S  
\* CA<21:0> ADDRESS LINES ALL 0'S  
\* AMR<15:0> FLOATING 1 PATTERN  
\* FOR EACH FLOATING 1 PATTERN AM BIT SHOULD READ 0.  
\*\*\*\*\*

1661 007274 000004  
007276 007306  
007300 060006  
007302 000000  
007304 060034  
007306 012737 001015 177746  
007314 004437 002342  
007320 007466

TST64:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO LOW CACHE SPACE

1662 007322 012737 000001 002066  
1663 007330 013737 002066 177752  
1664 007336 105037 177751  
1665 007342 105037 177750  
1666 007346 005737 000000  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676 007352 013703 177750  
1677 007356 000240  
007360 000240  
1678 007362 032703 000010  
1679 007366 001432  
1680 007370 005037 047314  
1681 007374 005037 047312  
1682 007400 013737 002066 047320  
1683  
1684 007406 013737 002066 047316  
1685 007414 012737 000017 002062  
1686 007422 006237 047316  
1687 007426 042737 100000 047316  
1688 007434 005337 002062  
1689 007440 001370  
1690 007442 104406

1\$: MOV #1,CHRPAT ;SETUP 1ST PATTERN FOR CHR<15:0>  
MOV CHRPAT,CHR ;LOAD AMR<15:0> FROM CHR<15:0>  
CLRB CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>  
CLRB CMR ;PRECONDITION AM BIT TO 0  
TST 0 ;PLACE ALL 0'S ON CA<21:0>.  
;WHEN PAX ADDRESS LINES ARE NOT BEING  
;ACCESSED BY THE CPU, THE CACHE DEFAULTS  
;TO SELECTING MA<21:0> ADDRESS LINES.  
;IN THIS SITUATION, MA<21:0> DEFAULTS  
;TO ALL 1'S THEREBY PLACING ALL 0'S  
;ON CA<21:0>. THEREFORE, FOLLOWING THE LOADING  
;OF ALL 0'S INTO AMR<21:0>, AND BEFORE THE  
;'TST 0' INSTRUCTION, ALL 0'S ARE PLACED  
; ON CA<21:0> ADDRESS LINES.  
25\$: MOV CMR,R3 ;SAVE AM BIT RESULT IN CMR  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
BIT #AM,R3 ;CHECK FOR 0.  
BEQ 9\$ ;PASS  
CLR CA210+2 ;PREPARE CA210 FOR TYPEOUT  
CLR CA210  
MOV CHRPAT,AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>  
;FOR ERROR TYPEOUT  
3\$: MOV CHRPAT,AMR210  
MOV #15,LOOP  
ASR AMR210  
BIC #100000,AMR210  
DEC LOOP  
BNE 3\$  
ERROR ;ERROR  
;-----

1691 007444 007442  
1692  
1693

.WORD -2  
;AMR TESTS  
;AM BIT SHOULD HAVE READ 0 INDICATING A  
;NO-MATCH CONDITION.

1694	007446	047312			CA210		:PRINT PATTERN USED FOR CACHE ADDRESS LINES CA<21:0>
1695	007450	047316			AMR210		:PRINT FLOATING 1 PATTERN USED FOR AMR<21:0> DATA
1696	007452	000000			.WORD	0	
1697	007454	006337	002066	9\$:	ASL	CHRPAT	:NEXT FLOATING 1 PATTERN
1698	007460	103401			BCS	10\$	:IF PHYSICAL ADDRESS 10000 DONE; THEN FINISHED
1699	007462	000722			BR	1\$	:IF NOT CONTINUE WITH NXT PATTERN
1700	007464	000240		10\$:	NOP		:END OF TEST
	007466	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

1707

```
.SBTTL TEST # 65 - FLOATING BIT TEST OF AM
*****
*TEST 65 - FLOATING BIT TEST OF AM
* MA<21:0> ADDRESS LINES ALL 1'S
* CA<21:0> ADDRESS LINES ALL 0'S
* AMR<21:16> FLOATING 1 PATTERN
* FOR EACH FLOATING 1 PATTERN AM BIT SHOULD READ 0.
*****
```

1708 007472 000004  
 007474 007504  
 007476 060006  
 007500 000000  
 007502 060034  
 007504 012737  
 007512 004437  
 007516 007646

```
TST65:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
.WORD 40$ ;TEST START LOCATION
.WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
.WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE
```

1709 007520 012737 000004 002064  
 1710 007526 113737 002064 177751  
 1711 007534 005037 177752  
 1712 007540 105037 177750  
 1713 007544 005737 000000  
 1714  
 1715  
 1716 007550 013703 177750  
 1717 007554 000240  
 007556 000240  
 1718 007560 032703 000010  
 1719 007564 001420  
 1720 007566 005037 047314  
 1721 007572 005037 047312  
 1722 007576 005037 047320  
 1723 007602 013737 002064 047316  
 1724 007610 006237 047316  
 1725 007614 104406

```
MOV #4,CMRPAT ;SETUP 1ST PATTERN FOR CMR<21:16>
MOVSB CMRPAT,CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>
CLR CHR ;LOAD ALL 0'S TO AMR<15:0> FROM CHR<15:0>
CLRB CMR ;PRECONDITION AM BIT TO 0
TST 0 ;SAVE CMR CONTENTS. BEFORE THE FETCH
;OF THIS INSTRUCTION, ALL 0'S WILL
;BE PLACED ON CA<21:0> LINES.
MOV CMR,R3 ;SAVE CMR CONTENTS
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIT #AM,R3 ;CHECK FOR 0.
BEQ 9$ ;PASS
CLR CA210+2 ;PREPARE CA210 FOR PRINTOUT
CLR CA210
CLR AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>
MOV CMRPAT,AMR210
ASR AMR210
ERROR ;ERROR
;-----
```

007616 007614  
 1726  
 1727  
 1728  
 1729 007620 047312  
 1730 007622 047316  
 1731 007624 000000  
 1732 007626 006337 002064  
 1733 007632 032737 000400 002064  
 1734 007640 001001  
 1735 007642 000731  
 1736 007644 000240  
 007646 005237 001472

```
.WORD -2 ;AMR TESTS
;AM BIT SHOULD HAVE READ 0 INDICATING A
;NO-MATCH CONDITION.
CA210 ;PRINT PATTERN USED FOR CACHE ADDRESS LINES CA<21:0>
AMR210 ;PRINT FLOATING 1 PATTERN USED FOR AMR<21:0> DATA
.WORD 0
ASL CMRPAT ;NEXT FLOATING 1 PATTERN
BIT #400,CMRPAT ;IF PHYSICAL ADDRESS 10000000 DONE;FINISHED
BNE 10$
BR 1$ ;IF NOT CONTINUE WITH NXT PATTERN
NOP ;END OF TEST
INC $TESTN ;INCREMENT TEST COUNTER
```

N 6  
 .SBTTL TEST # 66 - VERIFY NO MA TO CA LINES ARE SHORT TO EACH OTHER  
 \*\*\*\*\*  
 \*TEST 66 - VERIFY NO MA TO CA LINES ARE SHORT TO EACH OTHER  
 \* MA<12:0> ADDRESS LINES FLOATING 0  
 \* CA<12:0> ADDRESS LINES FLOATING 1  
 \* AMR<12:0> FLOATING 1 PATTERN  
 \* AM BIT READS 1  
 \*\*\*\*\*

1744 007652 000004  
 007654 007664  
 007656 060020  
 007660 000000  
 007662 060114  
 007664 012737 001015 177746 40\$:  
 007672 004437 002342  
 007676 010112

TST66: SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 :ERROR/LOOP ON TEST  
 :TEST START LOCATION  
 :LOOP ON ERROR START LOCATION  
 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 :LOOP ON ERROR END LOCATION  
 :DISABLE CACHE  
 :LOCATE TEST CODE TO LOW CACHE SPACE  
 :ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
 :ARE RELOCATED TO LOW CACHE SPACE

1745 007700 012737 000001 047344  
 1746 007706 012701 100001  
 1747  
 1748 007712 012737 170000 172350  
 1749  
 1750  
 1751  
 1752  
 1753 007720 012737 000001 177572 1\$:  
 1754 007726 012737 000020 172516  
 1755 007734 013737 047344 177752  
 1756 007742 105037 177751  
 1757 007746 105037 177750  
 1758 007752 023727 047344 000001  
 1759 007760 001004  
 1760 007762 105711  
 1761 007764 000240  
 1762 007766 000240  
 1763 007770 000403  
 1764 007772 005711 2\$:  
 1765 007774 000240  
 1766 007776 000240  
 1767 010000 013703 177750 4\$:  
 1768 010004 005037 177572  
 1769 010010 005037 172516  
 1770 010014 000240 25\$:  
 010016 000240  
 1771 010020 032703 000010  
 1772 010024 001017  
 1773 010026 013737 047344 047320  
 1774  
 1775 010034 005037 047316  
 1776 010040 013737 047344 047314  
 1777 010046 005037 047312  
 1778 010052 104406

MOV #1,FLTPAT :1ST FLOATING 1 PATTERN: 00001  
 MOV #100001,R1 :LOAD VIRTUAL ADDRESS. SELECTS KPAR4  
 :AND SPECIFIES OFFSET FOR PHYSICAL ADDRESS.  
 MOV #170000,KPAR4 :MAP PAGE 4 FOR TOP 124K ADDRESSING.  
 :TOGETHER WITH VIRTUAL ADDRESS WILL  
 :PLACE 17000001 ON PA LINES AND  
 :00000001 ON MA LINES FOR 1ST FLOATING  
 :1 PATTERN.  
 MOV #1,SRO :ENABLE MEM. MNGMENT.  
 MOV #20,SR3 :ENABLE 22-BIT MAPPING  
 MOV FLTPAT,CHR :LOAD AMR WITH FLOATING 1 PATTERN  
 CLRB CMR+1  
 CLRB CMR :PRECONDITION AM BIT TO 0  
 CMP FLTPAT,#1 :FOR 1ST PATTERN USE TSTB  
 BNE 2\$  
 TSTB (R1) :  
 NOP  
 NOP  
 BR 4\$  
 TST (R1) :  
 NOP  
 NOP  
 MOV CMR,R3 :DISABLE MEM. MNGMNT.  
 CLR SRO  
 CLR SR3  
 NOP :INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP :FOR LOOP ON ERROR  
 BIT #AM,R3 :CHECK FOR 1  
 BNE 9\$ :PASS  
 MOV FLTPAT,AMR210+2 :PREPARE PATTERN USED FOR AMR<21:0>  
 :FOR ERROR TYPEOUT  
 CLR AMR210  
 MOV FLTPAT,CA210+2 :PREPARE PATT. USED FOR CA<21:0 > FOR ERROR TYPE  
 CLR CA210  
 ERROR :ERROR



```

1779 010054 010052 .WORD .-2 ;-----
1780 ;AMR TESTS
1781 010056 047312 CA210 ;AM BIT DID NOT READ 1
1782 010060 047316 AMR210 ;PRINT FLOATING 1 PATTERN USED FOR CA<21:0>
1783 010062 000000 .WORD 0 ;PRINT FLOAT 1 PATTRN. USED FOR AMR<21:0>
1784 010064 006337 047344 9$: ASL FLTPAT ;NXT PATTERN
1785 010070 032737 020000 047344 BIT #20000,FLTPAT ;IF PATTERN 10000 DONE; FINISHED
1786 010076 001004 BNE 10$
1787 010100 006301 ASL R1 ;IF NOT NEXT PASS
1788 010102 052701 100000 BIS #100000,R1
1789 010106 000704 BR 1$
1790 010110 000240 10$: NOP ;END OF TEST
010112 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
    
```

```
.SBTTL TEST # 67 - AM FLOATING PATTERN TEST
:*****
:*TEST 67 - AM FLOATING PATTERN TEST
:* MA<21:13> ADDRESS LINES FLOATING 0
:* CA<21:13> ADDRESS LINES FLOATING 1
:* AMR<21:13> FLOAT. 1 PATRN.
:* AM BIT READS 1
:*****
```

```
1798 010116 000004
      010120 010130
      010122 060036
      010124 000000
      010126 060144
      010130 012737 001015 177746
      010136 004437 002342
      010142 010470
```

```
TST67:
      SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      .WORD 40$ ;ERROR/LOOP ON TEST
      .WORD 1$-40$+57764 ;TEST START LOCATION
      .WORD 0 ;LOOP ON ERROR START LOCATION
      .WORD 25$-40$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
      JSR R4,RELCTL ;DISABLE CACHE
      .WORD 10$+2 ;LOCATE TEST CODE TO LOW CACHE SPACE
      ;ADDRESS OF START OF NEXT TEST
```

```
:THE FOLLOWING LOCATIONS INCLUDING 10$
:ARE RELOCATED TO LOW CACHE SPACE
```

```
1799 010144 012737 171000 172350 2$:
1800
1801
1802 010152 012737 000040 047344
1803 010160 012737 020000 170220
1804
1805
1806
1807
1808
1809
1810 010166 012737 000200 002064
1811 010174 012737 020000 002066
1812 010202 012737 060126 000004
1813 010210 012737 000340 000006
1814 010216 113737 047345 170222
1815 010224 113737 002065 177751
1816 010232 013737 002066 177752
1817 010240 012737 000001 177572
1818 010246 012737 000060 172516
1819 010254 105037 177750
1820 010260 005737 100000
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832 010264 000240
1833 010266 000240
```

```
      MOV #171000,KPAR4 ;MAP PAGE 4 FOR TOP 124K ADDRESSING
      ;WILL ALSO SELECT UNIBUS MAP REGISTER
      ;SET #4
      MOV #40,FLTPAT ;SETUP 1ST PATTERN FOR UMPRO9
      MOV #20000,UMPRO8 ;SETUP 1ST PATTERN FOR UMPRO8
      ;ACCESSING TOP 124K,AND ENABLING
      ;UNIBUS MAP, THE 1ST ADDRESS WILL BE
      ;CONSTRUCTED THRU THE PA<21:)>
      ;LINES AS 17020000 AND AS 00020000
      ;THRU THE MA<21:0> LINES. DUE TO TOP 124K
      ;ADDRESSING CA<21:0> WILL SELECT THE MA LINES.
      MOV #200,CMRPAT ;SETUP 1ST PATTERN FOR CMR<15:10>
      MOV #20000,CHRPAT ;SETUP 1ST PATTERN FOR CHR<15:0>
      MOV #3$-2$+60000,4
      MOV #340,6
      MOVB FLTPAT+1,UMPRO9 ;LOAD UPPER BITS OF UNIBUS MAP REGISTER
      MOVB CMRPAT+1,CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>
      MOV CHRPAT,CHR ;LOAD AMR<15:0> FROM CHR<15:0>
      MOV #1,SRO ;ENABLE MEM MNGMENT
      MOV #60,SR3 ;ENABLE UNIBUS MAP AND 22-BIT MAPPING
      CLRB CMR ;PRECONDITION AM BIT WITH 0
      TST 100000 ;TOP 124K ADDRESSING WILL BE DONE PLACING
      ;THE APPROPRIATE FLOATING 1 ADDRESS PATTERN
      ;ON CA<21:0>.HOWEVER, THIS IS NOT WHEN THE
      ;AM BIT IS SET: WHEN PAX ADDRESS LINES ARE
      ;NOT BEING ACCESSED BY THE CPU,THE CACHE
      ;DEFAULTS TO SELECTING MA<21:0> ADDRESS
      ;LINES. IN THIS SITUATION,MA<21:0> DEFAULTS TO
      ;WHATEVER ADDRESS PATTERN IS BEING SET UP
      ;VIA THE UNIBUS MAP.
      ;THEREFORE AFTER THE 'CLRB CMR' INSTRUCTION
      ;AND BEFORE 'TST 100000' THE AM BIT SHOULD
      ;BE SET
      NOP
      NOP
```

```

1834 010270 000401          BR      4$          ;NO TRAP
1835 010272 022626          3$:    CMP      (SP)+,(SP)+ ;
1836 010274 013703 177750  4$:    MOV      CMR,R3      ;SAVE CMR CONTENTS
1837 010300 005037 177572          CLR      SR0
1838 010304 005037 172516          CLR      SR3          ;DISABLE UNIBUS MAP
1839 010310 000240          25$:   NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
        010312 000240          NOP          ;FOR LOOP ON ERROR
1840 010314 032703 000010          BIT      #AM,R3      ;CHECK FOR 1
1841 010320 001040          BNE     9$          ;PASS
1842 010322 012737 000006 000004          MOV      #6,4
1843 010330 005037 000006          CLR      6
1844 010334 013737 002066 047320          MOV      CHRPAT,AMR210+2 ;PREPARE AMR210 AND CA210 FOR PRINTOUT
1845 010342 013737 002066 047314          MOV      CHRPAT,CA210+2
1846 010350 013737 047344 047316          MOV      FLTPAT,AMR210
1847 010356 013737 047344 047312          MOV      FLTPAT,CA210
1848 010364 012737 000007 002062          MOV      #7,LOOP
1849 010372 006237 047312          6$:    ASR      CA210
1850 010376 006237 047316          ASR      AMR210
1851 010402 005337 002062          DEC      LOOP
1852 010406 001371          BNE     6$
1853 010410 104406          ERROR          ;ERROR
        010412 010410          .WORD   -2          ;-----
1854          ;AMR TESTS
1855          ;AM BIT DIT NOT READ 1
1856 010414 047312          CA210          ;PRINT CA<21:0> PATTERN USED
1857 010416 047316          AMR210         ;PRINT AMR<21:0> PAT. USED
1858 010420 000000          0
1859 010422 006337 047344          9$:    ASL      FLTPAT          ;NEXT PATTERN FOR UMPRO1
1860 010426 032737 040000 047344          BIT      #40000,FLTPAT ;IF ADDRESS PATTERN 10000000 DON; FINISHED
1861 010434 001007          BNE     8$
1862 010436 006337 170220          ASL      UMPROB          ;NEXT PATTERN FOR UMPROB
1863 010442 006337 002064          ASL      CMRPAT          ;NEXT PATTERN FOR CMR<15:10>
1864 010446 006337 002066          ASL      CHRPAT          ;NEXT PATTERN FOR CHR<15:0>
1865 010452 000653          BR      1$
1866 010454 012737 000006 000004          8$:    MOV      #6,4          ;RESTORE VECTORS
1867 010462 005037 000006          CLR      6
1868 010466 000240          10$:   NOP          ;END OF TEST
        010470 005237 001472          INC      $TESTN        ;INCREMENT TEST COUNTER
    
```

E 7  
 .SBTTL TEST # 70 - VERIFY NO STUCK PA LINES OR SHORTED TO EACH OTHER  
 \*\*\*\*\*  
 \*TEST 70 - VERIFY NO STUCK PA LINES OR SHORTED TO EACH OTHER  
 \* PA<14:0> ADDRESS LINES FLOATING 1  
 \* CA<14:0> ADDRESS LINES FLOATING 1  
 \* AMR<14:0> FLOATING 1 PATTERN  
 \* AM BIT READS 1  
 \*\*\*\*\*  
 TST70:

1876 010474 000004  
 010476 010506  
 010500 060004  
 010502 000000  
 010504 060034  
 010506 012737 001015 177746 40\$:  
 010514 004437 002342  
 010520 010636

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 - ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
 MOV #OFF,CCR ;DISABLE CACHE  
 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO LOW CACHE SPACE

1877 010522 012701 000001  
 1878 010526 010137 177752  
 1879 010532 105037 177751  
 1880 010536 105037 177750  
 1881 010542 022701 000001  
 1882 010546 001002  
 1883 010550 105711  
 1884 010552 000401  
 1885 010554 005711  
 1886  
 1887  
 1888 010556  
 1889 010556 000240  
 010560 000240  
 1890 010562 032737 000910 177750  
 1891 010570 001015  
 1892 010572 010137 047320  
 1893  
 1894 010576 005037 047316  
 1895 010602 010137 047314  
 1896 010606 005037 047312  
 1897 010612 104406

1\$: MOV #1,R1 ;R1 CONTAINS 1ST FLOATING 1 PATTERN: 000001  
 MOV R1,CHR ;LOAD AMR<15:0> FROM CHR<15:0>  
 CLRB CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>  
 CLRB CMR ;PRECONDITION AM BIT TO 0  
 CMP #1,R1 ;IF PATTERN IS 000001 USE TSTB  
 BNE 2\$  
 TSTB (R1)  
 BR 3\$  
 2\$: TST (R1) ;READ ADDRESS SPECIFIED IN R1  
 ;WHICH WILL PLACE FLOATING 1 PATTERN ON ADDRESS LINES  
 ;PA WILL BE SELECTED TO FEED CA LINES.  
 3\$:  
 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP ;FOR LOOP ON ERROR  
 BIT #AM,CMR ;CHECK FOR 1  
 BNE 9\$ ;PASS  
 MOV R1,AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>  
 ;FOR ERROR TYPE.  
 CLR AMR210  
 MOV R1,CA210+2 ;PREPARE PATTREN USED FOR CA<21:0>  
 CLR CA210  
 ERROR ;ERROR  
 ;-----

1898 010614 010612  
 1899  
 1900  
 1901 010616 047312  
 1902  
 1903 010620 047316  
 1904 010622 000000  
 1905 010624 006301  
 1906 010626 032701 100000  
 1907 010632 001735  
 1908 010634 000240

.WORD -2 ;AMR TESTS  
 ;AM BIT SHOULD HAVE READ 1 INDICATING A  
 ;MATCH CONDITION.  
 CA210 ;PRINT FLOATING 1 PATTERN USED FOR  
 ;CACHE ADDRESS LINES CA<21:0>  
 AMR210 ;PRINT PATTERN USED FOR AMR<21:0> DATA  
 .WORD 0  
 9\$: ASL R1 ;NEXT FLOATING 1 PATTERN  
 BIT #100000,R1 ;IS ADDRESS PATTERN 40000 DONE?  
 BEQ 1\$ ;NO; CONTINUE  
 10\$: NOP ;END OF TEST

010636 005237 001472

INC    \$TESTN

; INCREMENT TEST COUNTER

.SBTTL TEST # 71 - AFTER EACH FLOATING 1 PAT. CHECK AM BIT READS 1  
 \*\*\*\*\*  
 \*TEST 71 - AFTER EACH FLOATING 1 PAT. CHECK AM BIT READS 1  
 \* PA<21:15> FLOATING 1 PATTERN  
 \* CA<21:15> FLOATING 1 PATTERN  
 \* AMR<21:15> FLOATING 1 PATTERN  
 \*\*\*\*\*

1915 010642 000004  
 010644 010654  
 010646 060022  
 010650 000000  
 010652 060116  
 010654 012737 001015 177746  
 010662 004437 002342  
 010666 011166

TST71:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
 MOV #OFF,CCR ;DISABLE CACHE  
 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO LOW CACHE SPACE

1916 010670 012737 000002 002064  
 1917 010676 012737 100000 002066  
 1918 010704 012737 001000 172350  
 1919  
 1920  
 1921  
 1922  
 1923  
 1924  
 1925 010712 012737 060104 000004  
 1926 010720 012737 000340 000006  
 1927 010726 113737 002064 177751  
 1928 010734 013737 002066 177752  
 1929 010742 012737 000001 177572  
 1930 010750 012737 000020 172516  
 1931 010756 105037 177750  
 1932 010762 005737 100000  
 1933  
 1934  
 1935 010766 000240  
 1936 010770 000240  
 1937 010772 000401  
 1938 010774 022626  
 1939 010776 005037 177572  
 1940 011002 005037 172516  
 1941 011006 000240  
 011010 000240  
 1942 011012 032737 000010 177750  
 1943 011020 001044  
 1944 011022 012737 000006 000004  
 1945 011030 005037 000006  
 1946 011034 005037 047320  
 1947  
 1948 011040 005037 047314  
 1949 011044 013737 172350 047316  
 1950 011052 013737 172350 047312

2\$: MOV #2,CMRPAT ;1ST PATTERN FOR CMR<15:10>  
 MOV #100000,CHRPAT ;1ST PATTERN FOR CHR<15:0>  
 MOV #1000,KPAR4 ;SETUP 1ST PATRN. FOR PAGE ADDRESS FIELD  
 ;KPAR4 CONTAINS THE FLOATING 1  
 ;PATTERN AND REPRESENTS THE PAGE ADDRESS FIELD  
 ;DATA USED BY MEM. MNGMNT. TO  
 ;CONSTRUCT THE PHYSICAL ADDRESS.  
 ;1000 IS THE 1ST FLOATING 1 PATTERN  
 ;WHICH WILL BE CONSTRUCTED AS PHYS. ADDRESS 100000.  
 ;ALLOW FOR NEX TRAP  
 1\$: MOV #6\$-2\$+60000,4  
 MOV #340,6  
 MOVB CMRPAT,CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>  
 MOV CHRPAT,CHR ;LOAD AMR<15:0> FROM CHR<15:0>  
 MOV #1,SRO ;ENABLE MEM. MNGMNT.  
 MOV #20,SR3 ;ENABLE 22-BIT MAPPING  
 CLRB CMR ;PRECONDITION AM BIT TO 0  
 TST 100000 ;WILL CHOOSE KPAR4 FOR ADDRESSING.  
 ;PHYSICAL ADDRESS WILL BE DETERMINED  
 ;BY FLOATING PATTERN USED.  
 6\$: BR 7\$ ;NO TRAP  
 7\$: CMP (SP)+,(SP)+ ;ADJUST STACK  
 CLR SRO ;DISABLE MEM. MNGMNT.  
 CLR SR3  
 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP ;FOR LOOP ON ERROR  
 BIT #AM,CMR ;CHECK FOR 1  
 BNE 9\$ ;PASS  
 MOV #6,4  
 CLR 6  
 CLR AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>  
 ;AND CA<21:0> FOR TYPEOUT  
 CLR CA210+2  
 MOV KPAR4,AMR210  
 MOV KPAR4,CA210

1951	011060	012737	000011	002062		MOV	#9, LOOP	
1952	011066	006237	047316		5\$:	ASR	AMR210	
1953	011072	006237	047312			ASR	CA210	
1954	011076	042737	100000	047316		BIC	#100000, AMR210	
1955	011104	042737	100000	047312		BIC	#100000, CA210	
1956	011112	005337	002062			DEC	LOOP	
1957	011116	001363				BNE	5\$	
1958	011120	104406				ERROR		:ERROR
								:-----
	011122	011120				.WORD	.-2	
1959								:AMR TESTS
1960	011124	047312				CA210		:PRINT FLOAT. ADDRESS PATTERN USED
1961								:FOR CA<21:0>
1962	011126	047316				AMR210		:PRINT PATTERN USED FOR AMR<21:0>
1963	011130	000000				.WORD	0	
1964	011132	006337	172350		9\$:	ASL	KPAR4	:NEXT FLOATING 1 PATTERN
1965	011136	103405				BCS	8\$	:IF PHYSICAL ADDRESS 10000000 DONE; FINISHED
1966	011140	006337	002064			ASL	CMRPAT	:NEXT CMR PATTERN
1967	011144	006337	002066			ASL	CHRPAT	:NEXT CHR PATTERN
1968	011150	000660				BR	1\$	:CONTINUE
1969	011152	012737	000006	000004	8\$:	MOV	#6,4	:RESTORE VECTORS
1970	011160	005037	000006			CLR	6	
1971	011164	000240			10\$:	NOP		:END OF TEST
	011166	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

1975

```
.SBTTL TEST # 72 - LOADING TAG STORE FROM ADDRESS MATCH REGISTERS
:*****
:*TEST 72 - LOADING TAG STORE FROM ADDRESS MATCH REGISTERS
:* ALL 0'S TO TAG STORE ADDRESS LOCATION 0000.
:*****
TST72:
```

1975 011172 000004

011174 011204  
 011176 070000  
 011200 000000  
 011202 070042  
 011204 012737  
 011212 004437  
 011216 011354

001015 177746 40\$:  
 002370

```
SCPCND
.WORD 40$
.WORD 1$-40$+67764
.WORD 0
.WORD 25$-40$+67764
MOV #OFF,CCR
JSR R4,RELCTH
.WORD 10$+2
```

```
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;DISABLE CACHE
;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

1977 011220 005037 177752  
 1978 011224 112737 000003 177750

1\$: CLR CHR  
 MOV #HODO+TDAR,CMR

```
;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
;ALLOWS CACHE TAG FIELD BITS TO BE
;WRITTEN TO CHR<15:07> ONLY DURING
;THE DESTINATION MEMORY ACCESS
;OF AN INSTRUCTION
;ENABLE CACHE TAG FIELD TO BE WRITTEN
;INTO FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE TAG STORE
```

1979  
 1980  
 1981  
 1982  
 1983

1984 011232 012737 000015 177746  
 1985 011240 005737 040000  
 1986 011244 005737 060000  
 1987 011250 005737 060000

```
MOV #15,CCR
TST 40000
TST 60000
TST 60000
```

```
;WRITE INTO TAG STORE
;WRITE TAG FIELD DATA FROM CACHE ADDRESS
;LOCATION 0000 INTO CHR.
```

1988  
 1989 011254 013737 177752 047322  
 1990 011262 000240  
 011264 000240

25\$: MOV CHR,CHR157  
 NOP  
 NOP  
 CLRB CMR

```
;SAVE CHR DATA
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;DISABLE MAINTENANCE MODE
```

1991 011266 105037 177750  
 1992 011272 012737 001015 177746  
 1993 011300 042737 000177 047322  
 1994 011306 005737 047322

```
MOV #OFF,CCR
BIC #177,CHR157
TST CHR157
BEQ 10$
```

```
;PREPARE CHR157 FOR ERROR CHECK
;BITS <15:07> SHOULD BE ALL 0'S
;PASS
```

1995 011312 001417  
 1996 011314 012737 000007 002062  
 1997 011322 006237 047322  
 1998 011326 042737 100000 047322  
 1999 011334 005337 002062

2\$: MOV #7,LOOP  
 ASR CHR157  
 BIC #100000,CHR157  
 DEC LOOP  
 BNE 2\$

```
;ERROR;PREPARE CHR157 FOR TYPEOUT
```

2000 011340 001370  
 2001 011342 104406

ERROR

```
;ERROR
;-----
```

011344 011342

```
.WORD -2
```

```
;TAG STORE DATA TESTS
;READING TAGD<21:13> THRU CHR<15:07>
;DID NOT RESULT IN ALL 0'S.
;PRINT CHR<15:07>
```

2002  
 2003  
 2004  
 2005 011346 047322  
 2006 011350 000000  
 2007 011352 000240  
 011354 005237 001472

10\$: CHR157  
 .WORD 0  
 NOP  
 INC \$TESTN

```
;END OF TEST
;INCREMENT TEST COUNTER
```



2011

.SBTTL TEST # 73 - ALL 1'S TO TAG STORE ADDRESS LOCATION 0000  
 :\*\*\*\*\*  
 :\*TEST 73 - ALL 1'S TO TAG STORE ADDRESS LOCATION 0000  
 :\* ALL 1'S TO TAG STORE ADDRESS LOCATION 0000  
 :\*\*\*\*\*

2012 011360 000004

TST73: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE  
 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

2013 011406 012737 177777 177752 1\$:  
 2014 011414 112737 000003 177750

MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>  
 MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE  
 ;WRITTEN TO CHR<15:07> ONLY DURING  
 ;THE DESTINATION MEMORY ACCESS  
 ;OF AN INSTRUCTION  
 ;ENABLE CACHE TAG FIELD TO BE WRITTEN  
 ;INTO FROM AMR<8:0>  
 ;NO UCB SO AS TO WRITE ENABLE

2015  
 2016  
 2017  
 2018  
 2019  
 2020 011422 012737 000015 177746  
 2021 011430 005737 040000  
 2022 011434 005737 060000  
 2023 011440 005737 060000  
 2024

MOV #15,CCR  
 TST 40000  
 TST 60000 ;WRITE INTO TAG STORE  
 TST 60000 ;WRITE TAG FIELD DATA FROM CACHE ADDRESS  
 ;LOCATION 0000 INTO CHR.  
 ;SAVE CHR DATA  
 ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 ;FOR LOOP ON ERROR  
 ;DISABLE MAINTENANCE MODE

2025 011444 013737 177752 047322  
 2026 011452 000240  
 011454 000240

25\$: MOV CHR,CHR157  
 NOP  
 NOP  
 CLRB CMR  
 MOV #OFF,CCR  
 BIC #177,CHR157  
 CMP #177600,CHR157 ;BITS <15:07> SHOULD BE ALL 1'S  
 BEQ 10\$ ;PASS  
 MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT

2027 011456 105037 177750  
 2028 011462 012737 001015 177746  
 2029 011470 042737 000177 047322  
 2030 011476 022737 177600 047322  
 2031 011504 001417  
 2032 011506 012737 000007 002062  
 2033 011514 006237 047322  
 2034 011520 042737 100000 047322  
 2035 011526 005337 002062  
 2036 011532 001370  
 2037 011534 104406

28\$: ASR CHR157  
 BIC #100000,CHR157  
 DEC LOOP  
 BNE 2\$  
 ERROR ;ERROR  
 :-----

011536 011534

.WORD -2 ;TAG STORE DATA TESTS  
 ;READING TAGD<21:13> THRU CHR<15:07>  
 ;DID NOT RESULT IN ALL 1'S.  
 ;PRINT CHR<15:07>

2038  
 2039  
 2040  
 2041 011540 047322  
 2042 011542 000000  
 2043 011544 000240  
 011546 005237 001472

10\$: CHR157  
 .WORD 0  
 NOP  
 INC \$TESTN ;END OF TEST  
 ;INCREMENT TEST COUNTER

K 7  
 .SBTTL TEST # 74 - FLOAT 1 ACROSS 0'S TO TAG STORE ADRS LOC 0  
 \*\*\*\*\*  
 \*TEST 74 - FLOAT 1 ACROSS 0'S TO TAG STORE ADRS LOC 0  
 \* FLOAT 1 ACROSS 0'S TO TAG STORE ADDRESS LOCATION 0000  
 \*\*\*\*\*  
 TST74:

```

2048 011552 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          .WORD 40$          ;ERROR/LOOP ON TEST
011554 011564          .WORD 1$-40$+67764 ;TEST START LOCATION
011556 070006          .WORD 0          ;LOOP ON ERROR START LOCATION
011560 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
011562 070052          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
011564 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
011572 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
011576 011756          .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
    
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

```

2049 011600 012737 000001 047326          MOV #1,CHR80 ;1ST FLOATING 1 PATTERN:001
2050 011606 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2051 011614 013737 047326 177752          MOV CHR80,CHR ;LOAD AMR<8:0> VIA CHR<8:0> WITH
2052                                     ;FLOATING 1 PATTERN
2053 011622 112737 000003 177750          MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2054                                     ;WRITTEN TO CHR<15:07> ONLY DURING
2055                                     ;THE DESTINATION MEMORY ACCESS
2056                                     ;OF AN INSTRUCTION
2057                                     ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2058                                     ;INTO FROM AMR<8:0>
2059 011630 005737 040000          TST 40000
2060 011634 005737 060000          TST 60000 ;WRITE INTO TAG STORE
2061 011640 005737 060000          TST 60000 ;WRITE TAG FIELD DATA FROM CACHE ADDRESS
2062                                     ;LOCATION 0000 INTO CHR.
2063 011644 013737 177752 047322          MOV CHR,CHR157 ;SAVE CHR DATA
2064 011652 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
          011654 000240          NOP ;FOR LOOP ON ERROR
2065 011656 105037 177750          CLRB CMR ;DISABLE MAINTENANCE MODE
2066 011662 012737 001015 177746          MOV #OFF,CCR ;DISABLE CACHE
2067 011670 012737 000007 002062          MOV #7,LOOP ;PREPARE CHR157 FOR COMPARISON
2068 011676 006237 047322 3$: ASR CHR157
2069 011702 042737 100000 047322          BIC #100000,CHR157
2070 011710 005337 002062          DEC LOOP
2071 011714 001370          BNE 3$
2072 011716 023737 047326 047322          CMP CHR80,CHR157 ;CHECK FOR CORRECT PATTERN
2073 011724 001405          BEQ 9$ ;PASS
2074 011726 104406          ERROR ;ERROR
          011730 011726          .WORD -2 ;-----
2075                                     ;TAG STORE DATA TESTS
2076                                     ;READING CHR<15:07> FOR TAGD<21:13>
2077                                     ;DID NOT RESULT IN CORRECT FLOATING
2078                                     ;1 PATTERN.
2079 011732 047322          CHR157 ;PRINT CHR<15:07>
2080 011734 047326          CHR80 ;PRINT FLOATING 1 PATTERN LOADED
2081                                     ;INTO CHR<8:0>
2082 011736 000000          .WORD 0
2083 011740 006337 047326 9$: ASL CHR80 ;NEXT PATTERN
    
```

2084	011744	032737	001000	047326		BIT	#1000,CHR80	:IF PATTERN 400 DONE;FINISHED
2085	011752	001715				BEQ	1\$	:IF NOT, NEXT PASS
2086	011754	000240			10\$:	NOP		:END OF TEST
	011756	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

M 7  
 .SBTTL TEST # 75 - CHECK ALL LOW CACHE TAG STORE ADRS LOCS  
 \*\*\*\*\*  
 \*TEST 75 - CHECK ALL LOW CACHE TAG STORE ADRS LOCS  
 \* WRITE AND READ 0'S TO ALL LOW CACHE TAG STORE ADDRESS LOCATIONS  
 \*\*\*\*\*  
 TST75:

```

2091 011762 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          011764 011774          .WORD 40$          ;ERROR/LOOP ON TEST
          011766 070014          .WORD 1$-40$+67764 ;TEST START LOCATION
          011770 000000          .WORD 0          ;LOOP ON ERROR START LOCATION
          011772 070044          .WORD 25$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          011774 012737 001015 177746 40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
          012002 004437 002370 JSR R4,RELCTH ;DISABLE CACHE
          012006 012176          .WORD 10$+2 ;LOCATE TEST CODE TO HIGH CACHE SPACE
  
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

```

2092 012010 005037 177752          CLR CHR          ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
2093 012014 012705 060000          MOV #60000,R5   ;ADDRESS 60000 INTO R5
2094 012020 012703 040000          MOV #40000,R3   ;ADDRESS 40000 INTO R3
2095 012024 012737 000015 177746 1$: MOV #15,CCR     ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2096 012032 112737 000003 177750 MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2097                                     ;WRITTEN TO CHR<15:07> ONLY DURING
2098                                     ;THE DESTINATION MEMORY ACCESS
2099                                     ;OF AN INSTRUCTION
2100                                     ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2101                                     ;INTO FROM AMR<8:0>
2102 012040 005713          TST (R3)        ;
2103 012042 005715          TST (R5)        ;WRITE INTO TAG STORE
2104 012044 005715          TST (R5)        ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2105                                     ;LOCATION SPECIFIED BY CA<12:1> IN R5.
2106 012046 013737 177752 047322 MOV CHR,CHR157 ;SAVE CHR DATA
2107 012054 000240          NOP             ;INSTRUCTION 'JMP 1$' PLACED HERE
          012056 000240          NOP             ;FOR LOOP ON ERROR
2108 012060 105037 177750          CLRB CMR       ;DISABLE MAINTENANCE MODE
2109 012064 012737 001015 177746 MOV #OFF,CCR    ;DISABLE CACHE
2110 012072 042737 000177 047322 BIC #177,CHR157 ;PREPARE CHR157 FOR ERROR CHECK
2111 012100 005737 047322          TST CHR157    ;BITS <15:07> SHOULD BE ALL 0'S
2112 012104 001424          BEQ 9$         ;PASS
2113 012106 010537 047314          MOV R5,CA210+2 ;SAVE CACHE ADDRESS USED: CA<21:0>
2114 012112 005037 047312          CLR CA210
2115 012116 012737 000007 002062 MOV #7,LOOP     ;ERROR:PREPARE CHR157 FOR TYPEOUT
2116 012124 006237 047322          ASR CHR157
2117 012130 042737 100000 047322 BIC #100000,CHR157
2118 012136 005337 002062          DEC LOOP
2119 012142 001370          BNE 4$
2120 012144 104406          ERROR          ;ERROR
  
```

```

          012146 012144          .WORD -2
2121                                     ;TAG STORE DATA TESTS
2122                                     ;READING TAGD<21:13> THRU CHR<15:07>
2123                                     ;DID NOT RESULT IN ALL 0'S.
2124 012150 047322          CHR157          ;PRINT CHR<15:07>
2125 012152 047312          CA210           ;PRINT CA<21:0> ADDRESS USED
2126                                     ;BITS <12:1> IS THE CACHE TAG STORE ADDRESS
  
```

2127  
 2128 012154 000000  
 2129 012156 062705 000002  
 2130 012162 062703 000002  
 2131 012166 020527 070000  
 2132 012172 001314  
 2133 012174 000240  
       012176 005237 001472

9\$:      .WORD      0  
           ADD        #2,R5  
           ADD        #2,R3  
           CMP        R5,#70000  
           BNE        1\$  
 10\$:     NOP  
           INC        \$TESTN

:LOCATION FAILURE  
 :NEXT CACHE STORE LOCATION  
 :HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN CHECKED?  
 :NO  
 :END OF TEST  
 :INCREMENT TEST COUNTER

SBTTL TEST # 76 - TEST ALL LOW CACHE TAG STORE LOCATIONS  
 \*\*\*\*\*  
 \*TEST 76 - TEST ALL LOW CACHE TAG STORE LOCATIONS  
 \* WRITE AND READ 1'S TO ALL LOW CACHE TAG STORE ADDRESS LOCATIONS  
 \* (0000 TO 37777)  
 \*\*\*\*\*

```

TST76:
2139 012202 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
012204 012214          .WORD 40$
012206 070016          .WORD 1$-40$+67764
012210 000000          .WORD 0
012212 070046          .WORD 25$-40$+67764
012214 012737 001015 177746 40$: MOV #OFF,CCR
012222 004437 002370 JSR R4,RELCTH
012226 012422          .WORD 10$+2

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

2140 012230 012737 177777 177752          MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
2141 012236 012705 060000          MOV #60000,R5 ;ADDRESS 60000 INTO R5
2142 012242 012703 040000          MOV #40000,R3 ;ADDRESS 40000 INTO R3
2143 012246 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2144 012254 112737 000003 177750          MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2145          ;WRITTEN TO CHR<15:07> ONLY DURING
2146          ;THE DESTINATION MEMORY ACCESS
2147          ;OF AN INSTRUCTION
2148          ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2149          ;INTO FROM AMR<8:0>
2150 012262 005713          TST (R3)
2151 012264 005715          TST (R5) ;WRITE INTO TAG STORE
2152 012266 005715          TST (R5) ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2153          ;LOCATION SPECIFIED BY CA<12:1> IN R5
2154 012270 013737 177752 047322          MOV CHR,CHR157 ;SAVE CHR DATA
2155 012276 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
012300 000240          NOP ;FOR LOOP ON ERROR
2156 012302 105037 177750          CLRB CMR ;DISABLE MAINTENANCE
2157 012306 012737 001015 177746          MOV #OFF,CCR
2158 012314 042737 000177 047322          BIC #177,CHR157
2159 012322 022737 177600 047322          CMP #177600,CHR157 ;BITS <15:07> SHOULD BE ALL 1'S
2160 012330 001424          BEQ 9$ ;PASS
2161 012332 010537 047314          MOV R5,CA210+2 ;SAVE CACHE ADDRESS USED: CA<21:0>
2162 012336 005037 047312          CLR CA210
2163 012342 012737 000007 002062          MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
2164 012350 006237 047322 4$: ASR CHR157
2165 012354 042737 100000 047322          BIC #100000,CHR157
2166 012362 005337 002062          DEC LOOP
2167 012366 001370          BNE 4$
2168 012370 104406          ERROR ;ERROR
                                ;-----
012372 012370          .WORD -2

2169          ;TAG STORE DATA TESTS
2170          ;READING TAGD<21:13> THRU CHR<15:07>
2171          ;DID NOT RESULT IN ALL 1'S.
2172 012374 047312          CA210 ;PRINT CACHE ADDRESS CA<21:0>
2173 012376 047322          CHR157 ;PRINT CHR<15:07>
    
```

2174	012400	000000			.WORD	0	
2175	012402	062705	000002	9\$:	ADD	#2,R5	;NEXT CACH LOCATION
2176	012406	062703	000002		ADD	#2,R3	
2177	012412	020527	070000		CMP	R5,#70000	;HAVE ALL LOCATIONS BEEN DONE?
2178	012416	001313			BNE	1\$	;NO
2179	012420	000240		10\$:	NOP		;END OF TEST
	012422	005237	001472		INC	\$TESTN	;INCREMENT TEST COUNTER

TEST # 77 - TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS  
2184

.SBTTL TEST # 77 - TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS  
\*\*\*\*\*  
\*TEST 77 - TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS  
\* WRITE AND READ 0'S TO ALL HI CACHE TAG STORE ADDRESS LOCATIONS  
\* (4000 TO 7777)  
\*\*\*\*\*

2185 012426 000004  
012430 012440  
012432 060014  
012434 000000  
012436 060044  
012440 012737 001015 177746 40\$:  
012446 004437 002342  
012452 012642

TST77:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO LOW CACHE SPACE

2186 012454 005037 177752  
2187 012460 012705 070000  
2188 012464 012703 050000  
2189 012470 012737 000015 177746 1\$:  
2190 012476 112737 000003 177750  
2191  
2192  
2193  
2194  
2195  
2196 012504 005713  
2197 012506 005715  
2198 012510 005715  
2199  
2200 012512 013737 177752 047322  
2201 012520 000240 25\$:  
012522 000240  
2202 012524 105037 177750  
2203 012530 012737 001015 177746  
2204 012536 042737 000177 047322  
2205 012544 005737 047322  
2206 012550 001424  
2207 012552 010537 047314  
2208 012556 005037 047312  
2209 012562 012737 000007 002062  
2210 012570 006237 047322 4\$:  
2211 012574 042737 100000 047322  
2212 012602 005337 002062  
2213 012606 001370  
2214 012610 104406

CLR CHR ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>  
MOV #70000,R5 ;ADDRESS 70000 INTO R5  
MOV #50000,R3 ;ADDRESS 50000 INTO R3  
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
MOV #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE  
;WRITTEN TO CHR<15:07> ONLY DURING  
;THE DESTINATION MEMORY ACCESS  
;OF AN INSTRUCTION  
;ENABLE CACHE TAG FIELD TO BE WRITTEN  
;INTO FROM AMR<8:0>  
TST (R3)  
TST (R5) ;WRITE INTO TAG STORE  
TST (R5) ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS  
;LOCATION SPECIFIED BY CA<12:1> IN R5.  
MOV CHR,CHR157 ;SAVE CHR DATA  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINTENANCE MODE  
MOV #OFF,CCR ;DISABLE CACHE  
BIC #177,CHR157 ;PREPARE CHR157 FOR ERROR CHECK  
TST CHR157 ;BITS <15:07> SHOULD BE ALL 0'S  
BEQ 9\$ ;PASS  
MOV R5,CA210+2 ;SAVE CACHE ADDRESS USED: CA<21:0>  
CLR CA210  
MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT  
ASR CHR157  
BIC #100000,CHR157  
DEC LOOP  
BNE 4\$  
ERROR ;ERROR  
;-----

012612 012610  
2215  
2216  
2217  
2218 012614 047322  
2219 012616 047312

.WORD -2  
CHR157  
CA210  
;TAG STORE DATA TESTS  
;READING TAGD<21:13> THRU CHR<15:07>  
;DID NOT RESULT IN ALL 0'S.  
;PRINT CHR<15:07>  
;PRINT CA<21:0> ADDRESS USED



```
2220                                     ;BITS <12:1> IS THE CACHE TAG STORE ADDRESS
2221                                     ;LOCATION FAILURE
2222 012620 000000                                     ;NEXT CACHE STORE LOCATION
2223 012622 062705 000002    9$:   .WORD    0
2224 012626 062703 000002    ADD     #2,R5
2225 012632 020527 100000    ADD     #2,R3
2226 012636 001314                                     ;HAVE ALL HI CACHE ADDRESS LOCATIONS BEEN CHECKED?
2227 012640 000240                                     ;NO
      012642 005237 001472    10$:  CMP     R5,#100000
                                     ;END OF TEST
                                     ;INCREMENT TEST COUNTER
      INC     $TESTN
```

TEST # 100 - TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS

2232

.SBTTL TEST # 100 - TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS

\*\*\*\*\*
:TEST 100 - TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS
:\* WRITE AND READ 1'S TO ALL HI CACHE TAG STORE ADDRESS LOCATIONS
:\* (4000 TO 7777)
\*\*\*\*\*

TST100:

2233 012646 000004

SCPCND

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
:DISABLE CACHE
:LOCATE TEST CODE TO LOW CACHE SPACE
:ADDRESS OF START OF NEXT TEST

012650 012660
012652 060016
012654 000000
012656 060046
012660 012737 001015 177746
012666 004437 002342
012672 013066

.WORD 40\$
:WORD 1\$-40\$+57764
:WORD 0
:WORD 25\$-40\$+57764
MOV #OFF,CCR
JSR R4,RELCTL
:WORD 10\$+2

:THE FOLLOWING LOCATIONS INCLUDING 10\$
:ARE RELOCATED TO LOW CACHE SPACE

2234 012674 012737 177777 177752

MOV #-1,CHR
MOV #70000,R5
MOV #50000,R3
MOV #15,CCR
MOVB #HODO+TDAR,CMR

:LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
:ADDRESS 70000 INTO R5
:ADDRESS 50000 INTO R3
:NO UCB SO AS TO WRITE ENABLE CACHE STORE
:ALLOWS CACHE TAG FIELD BITS TO BE
:WRITTEN TO CHR<15:07> ONLY DURING
:THE DESTINATION MEMORY ACCESS
:OF AN INSTRUCTION
:ENABLE CACHE TAG FIELD TO BE WRITTEN
:INTO FROM AMR<8:0>

2235 012702 012705 070000
2236 012706 012703 050000
2237 012712 012737 000015 177746
2238 012720 112737 000003 177750
2239
2240
2241
2242
2243

TST (R3)
TST (R5)
TST (R5)

:WRITE INTO TAG STORE
:WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
:LOCATION SPECIFIED BY CA<12:1> IN R5
:SAVE CHR DATA
:INSTRUCTION 'JMP 1\$' PLACED HERE
:FOR LOOP ON ERROR
:DISABLE MAINTENANCE

2244 012726 005713
2245 012730 005715
2246 012732 005715
2247

2248 012734 013737 177752 047322
2249 012742 000240
012744 000240

25\$:

MOV CHR,CHR157
NOP
NOP
CLRB CMR
MOV #OFF,CCR
BIC #177,CHR157
CMP #177600,CHR157
BEQ 9\$
MOV R5,CA210+2
CLR CA210
MOV #7,LOOP
ASR CHR157
BIC #100000,CHR157
DEC LOOP
BNE 4\$
ERROR

:BITS <15:07> SHOULD BE ALL 1'S
:PASS
:SAVE CACHE ADDRESS USED: CA<21:0>
:ERROR;PREPARE CHR157 FOR TYPEOUT

2250 012746 105037 177750
2251 012752 012737 001015 177746
2252 012760 042737 000177 047322
2253 012766 022737 177600 047322
2254 012774 001424
2255 012776 010537 047314
2256 013002 005037 047312
2257 013006 012737 000007 002062
2258 013014 006237 047322
2259 013020 042737 100000 047322
2260 013026 005337 002062
2261 013032 001370
2262 013034 104406

4\$:

MOV #7,LOOP
ASR CHR157
BIC #100000,CHR157
DEC LOOP
BNE 4\$
ERROR

:ERROR
:-----

013036 013034

.WORD -2

2263
2264
2265

:TAG STORE DATA TESTS
:READING TAGD<21:13> THRU CHR<15:07>
:DID NOT RESULT IN ALL 1'S.
:PRINT CACHE ADDRESS CA<21:0>
:PRINT CHR<15:07>

2266 013040 047312
2267 013042 047322

CA210
CHR157

2268	013044	000000			.WORD	0	
2269	013046	062705	000002	9\$:	ADD	#2,R5	:NEXT CACH LOCATION
2270	013052	062703	000002		ADD	#2,R3	
2271	013056	020527	100000		CMP	R5,#100000	:HAVE ALL LOCATIONS BEEN DONE?
2272	013062	001313			BNE	1\$	:NO
2273	013064	000240		10\$:	NOP		:END OF TEST
	013066	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

2279

```
.SBTTL TEST # 101 - LOADING TAG STORE FROM CACHE ADRS LINES CA(21:13)
:*****
:*TEST 101 - LOADING TAG STORE FROM CACHE ADRS LINES CA(21:13)
:* CHECK LOADING OF TAG STORE DATA(TAG WRTD<21:13>) FROM
:* CACHE ADDRESS LINES CA<21:13>.
:* WRITE ALL 0'S IN TAG STORE LOCATION 0000 FROM CA<21:13>
:*****
```

2280 013072 000004  
 013074 013104  
 013076 070000  
 013100 000000  
 013102 070036  
 013104 012737 001015 177746  
 013112 004437 002370  
 013116 013250

```
TST101:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
40$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
;WORD 40$
;WORD 1$-40$+67764
;WORD 0
;WORD 25$-40$+67764
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

2281 013120 112737 000002 177750 1\$:  
 2282  
 2283  
 2284  
 2285 013126 012737 000015 177746  
 2286 013134 005737 040000  
 2287 013140 005737 000000  
 2288  
 2289 013144 005737 000000  
 2290  
 2291 013150 013737 177752 047322  
 2292 013156 000240 25\$:  
 013160 000240  
 2293 013162 105037 177750  
 2294 013166 012737 001015 177746  
 2295 013174 042737 000177 047322  
 2296 013202 005737 047322  
 2297 013206 001417  
 2298 013210 012737 000007 002062  
 2299 013216 006237 047322 2\$:  
 2300 013222 042737 100000 047322  
 2301 013230 005337 002062  
 2302 013234 001370  
 2303 013236 104406

```
MOVB #HODO,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
;WRITTEN TO CHR<15:07> ONLY DURING
;THE DESTINATION MEMORY ACCESS
;OF AN INSTRUCTION
;NO UCB SO AS TO WRITE ENABLE CACHE STORE
MOV #15,CCR
TST 40000
TST 0 ;WRITE ALL 0'S INTO TAG STORE LOCATION 0000
;FROM CACHE ADDRESS CA<21:13>
TST 0 ;WRITE TAG STORE DATA FROM LOCATION
;0000 INTO CHR<15:07>.
MOV CHR,CHR157 ;SAVE CHR DATA
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINTENANCE MODE
MOV #OFF,CCR ;DISABLE CACHE
BIC #177,CHR157 ;PREPARE CHR157 FOR ERROR CHECK
TST CHR157 ;BITS <15:07> SHOULD BE ALL 0'S
BEQ 10$ ;PASS
MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
2$: ASR CHR157
BIC #100000,CHR157
DEC LOOP
BNE 2$
ERROR ;ERROR
;-----
```

2304 013240 013236  
 2305  
 2306  
 2307 013242 047322  
 2308 013244 000000  
 2309 013246 000240 10\$:  
 013250 005237 001472

```
.WORD -2 ;TAG STORE DATA TESTS
;READING TAGD<21:13> THRU CHR<15:07>
;DID NOT RESULT IN ALL 0'S.
;PRINT CHR<15:07>
CHR157
;WORD 0
NOP ;END OF TEST
10$: INC $TESTN ;INCREMENT TEST COUNTER
```

TEST # 102 - WRITE FLOATING 1 ACROSS 0'S INTO TAG LOC 0

SBTTL TEST # 102 - WRITE FLOATING 1 ACROSS 0'S INTO TAG LOC 0  
 \*\*\*\*\*  
 \*TEST 102 - WRITE FLOATING 1 ACROSS 0'S INTO TAG LOC 0  
 \* WRITE FLOATING 1 ACROSS 0'S INTO TAG STORE LOCATION 0000  
 \* FROM CA<21:13> USING AVAILABLE MEMORY.  
 \* PROCEDURE: STARTING AT 8K BOUNDARY(ADDR. 20000) CHECK  
 \* FOR AVAILABLE FLOATING ADDRESS UP TO ADDR. 1000000  
 \* WHEN THE FLOATING ADDRESS EXISTS PERFORM THE TEST.  
 \*\*\*\*\*

2317  
 013254  
 2318 013254 000004  
 013256 013266  
 013260 070006  
 013262 000000  
 013264 070100  
 013266 012737 001015 177746  
 013274 004437 002370  
 013300 013566

TST102:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 18-40\$+67764 ;LOOP ON ERROR START LOCATION  
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
 MOV #OFF,CCR ;DISABLE CACHE  
 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

2319 013302 012737 000200 172350  
 2320  
 2321  
 2322  
 2323  
 2324  
 2325 013310 012737 070240 000004  
 2326 013316 012737 000340 000006  
 2327 013324 112737 000002 177750  
 2328  
 2329  
 2330 013332 012737 000001 177572  
 2331 013340 012737 000020 172516  
 2332 013346 012737 000015 177746  
 2333 013354 005737 040000  
 2334 013360 005737 100000  
 2335  
 2336  
 2337  
 2338 013364 000240  
 2339 013366 000240  
 2340 013370 005737 100000  
 2341  
 2342 013374 013737 177752 047322  
 2343 013402 000240  
 013404 000240  
 2344 013406 005037 177572  
 2345 013412 005037 172516  
 2346 013416 012737 001015 177746  
 2347 013424 105037 177750  
 2348 013430 042737 000177 047322  
 2349 013436 023737 172350 047322  
 2350 013444 001437  
 2351 013446 012737 000006 000004

2\$: MOV #200,KPAR4 ;KPAR4 CONTAINS THE FLOATING 1 PATTERN  
 ;AND REPRESENTS THE THE PAGE ADDRESS FIELD  
 ;DATA USED BY MEMORY MNGMNT. TO CONSTRUCT  
 ;THE PHYSICAL ADDRESS. 200 IS THE 1ST  
 ;FLOATING 1 PATTERN WHICH WILL BE CONSTRUCTED  
 ;TO ADDRESS 20000.  
 1\$: MOV #7\$-2\$+70000,4 ;ALLOW FOR NEX TRAP  
 MOV #340,6  
 MOVB #HODO,CMR ;ALLOWS CACHE TAG STORE TO BE WRITTEN  
 ;TO CHR<15:07> ONLY DURING THE DESTINATION  
 ;MEMORY ACCESS OF AN INSTRUCTION.  
 MOV #1,SRO ;ENABLE MEMORY MNGMNT.  
 MOV #20,SR3 ;ENABLE 22-BIT MAPPING  
 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
 TST 40000  
 TST 100000 ;CHOOSES KPAR4 FOR ADDRESSING. PHYSICAL  
 ;ADDRESS WILL BE DETERMINED BY FLOATING  
 ;PATTERN USED IN KPAR4. TAG STORE WILL  
 ;BE WRITTEN WITH DATA PLACED ON CA<21:13> ADDRESS LINES.  
 NOP  
 NOP ;NO TRAP  
 TST 100000 ;WRITE TAG STORE DATA FROM LOCATION  
 ;0000 INTO CHR<15:07>.  
 25\$: MOV CHR,CHR157 ;SAVE CHR INFO.  
 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP ;FOR LOOP ON ERROR  
 CLR SRO ;DISABLE MEM MNGMENT.  
 CLR SR3  
 MOV #OFF,CCR ;DISABLE CACHE  
 CLRB CMR ;DISABLE MAINTENANCE  
 BIC #177,CHR157  
 CMP KPAR4,CHR157 ;IS THERE ERROR?  
 BEQ 9\$ ;PASS  
 MOV #6,4

2352	013454	005037	000006		CLR	6		
2353	013460	013737	172350	047324	MOV	KPAR4,CA2113	:	SAVE PATTERN USED FOR CA<21:13>.
2354	013466	012737	000007	002062	MOV	#7,LOOP	:	PREPARE CHR157 AND CA2113 FOR ERROR PRINT
2355	013474	006237	047322		ASR	CHR157		
2356	013500	006237	047324		ASR	CA2113		
2357	013504	042737	100000	047322	BIC	#100000,CHR157		
2358	013512	042737	100000	047324	BIC	#100000,CA2113		
2359	013520	005337	002062		DEC	LOOP		
2360	013524	001363			BNE	5%		
2361	013526	104406			ERROR		:	ERROR
							:	-----
	013530	013526			.WORD	.-2		
2362							:	TAG STORE TESTS
2363							:	READING CHR<15:07> FOR TAG DATA (TAGD <21:13>)
2364							:	DID NOT RESULT IN CORRECT ADDRESS PATTERN
2365							:	LOADED FROM CA<21:13>.
2366	013532	047322			CHR157		:	PRINT CHR<15:07>
2367	013534	047324			CA2113		:	PRINT ADDRESS PATTERN USED: CA<21:13>
2368	013536	000000			.WORD	0	:	PRINT TERMINATE
2369	013540	000401			BR	9%	:	NEXT PATTERN
2370	013542	022626			CMP	(SP)+,(SP)+	:	RESTORE STACK DUE TO INTERRUPT
2371	013544	006337	172350		ASL	KPAR4	:	NEXT PATTERN;IF PHYSICAL ADDRESS
2372							:	1000000 HAS BEEN DONE; FINISHED
2373	013550	103257			BCC	1%		
2374	013552	012737	000006	000004	MOV	#6,4	:	RESTORE VECTORS
2375	013560	005037	000006		CLR	6		
2376	013564	000240			NOP		:	END OF TEST
	013566	005237	001472		INC	\$TESTN	:	INCREMENT TEST COUNTER

CKKKABO 11-44 KK11B CACHE MACRO M1113 27-OCT-80 08:39 PAGE 88 SEQUENCE 101  
TEST # 103 - FLOAT 1 ACROSS 0'S INTO TAG STORE ADRS LOC 0  
2387

.SBTTL TEST # 103 - FLOAT 1 ACROSS 0'S INTO TAG STORE ADRS LOC 0  
\*\*\*\*\*  
\*TEST 103 - FLOAT 1 ACROSS 0'S INTO TAG STORE ADRS LOC 0  
\*WRITE FLOATING 1 ACROSS 0'S INTO TAG STORE ADDRESS LOCATION 0000  
\*FROM CA<21:13> USING RMI REGISTER (G5179)  
\*PROCEDURE: START AT 16K BOUNDARY (ADDR. 100000) AND CHECK FOR  
\*AVAILABLE FLOATING ADDRESSES UP TO ADDR. 1000000  
\*WHEREVER A FLOATING ADDRESS DOES NOT EXIST USE  
\*THE RMI REGISTER. IF ADDRESS EXISTS DO NOT PERFORM  
\*THE TEST SINCE THAT LOCATION WOULD HAVE BEEN TESTED  
\*BY THE PREVIOUS TEST.  
\*\*\*\*\*

2388 013572 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
013574 013604 .WORD 40\$ ;ERROR/LOOP ON TEST  
013576 070050 .WORD 1\$-40\$+67764 ;TEST START LOCATION  
013600 000000 .WORD 0 ;LOOP ON ERROR START LOCATION  
013602 070166 .WORD 25\$-40\$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
013604 012737 001015 177746 40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION  
013612 004437 002370 JSR R4,RELCTH ;DISABLE CACHE  
013616 014344 .WORD 10\$+2 ;LOCATE TEST CODE TO HIGH CACHE SPACE  
;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

2389 013620 132737 000200 001507 2\$: BITB #APTSIZE,\$ENVM ;DOES APT SIZE?  
2390 013626 001405 BEQ 11\$ ;NO ,GO AUTOSIZE  
2391 013630 032737 000200 001512 BIT #200,\$USWR ;DOES APT INDICATE  
2392 ;THAT RMI REGISTER IS PRESENT  
2393 013636 001006 BNE 5\$ ;YES,USE IT  
2394 013640 000555 BR 4\$ ;APT SAYS DO NOT PERFORM TEST  
2395 013642 012737 070352 000004 11\$: MOV #3\$-2\$+70000,4 ;AUTO-SIZE FOR RMI,PREPARE FOR TRAP  
2396 013650 005737 177770 TST 177770 ;READ RMI  
2397 013654 012737 001000 172350 5\$: MOV #1000,KPAR4 ;SETUP MEM. MNG. PAGE 4 FOR FIRST FLO..TING  
2398 ;ADDRESS 100000  
2399 013662 012737 000002 047344 MOV #2,FLTPAT ;SETUP 1ST FLOATING PATTERN FOR RMI  
2400 ;REG. CORRESPONDING TO ADDRESS 40000  
2401 013670 012737 070100 000004 1\$: MOV #8\$-2\$+70000,4 ;SETUP FOR NEX MEMORY  
2402 013676 012737 000001 177572 MOV #1,SRO ;ENABLE MEM.MNGMENT.  
2403 013704 012737 000020 172516 MOV #20,SR3 ;ENABLE 22 BIT MAPPING  
2404 013712 005737 100000 TST 100000 ;SELECT PAGE 4. READ ADDRESS SPECIFIED BY KPAR4.  
2405 013716 000512 BR 9\$ ;NO TRAP.MEMORY LOCATION EXISTS,SO DON'T  
2406 ;BOTHER TESTING WITH RMI FOR THIS LOCATION  
2407 013720 022626 8\$: CMP (SP)+,(SP)+ ;TRAP HERE WHEN FLOATING ADDRESS  
2408 ;LOCATION DOES NOT EXIST.USE RMI FOR TESTING  
2409 013722 013701 047344 MOV FLTPAT,R1 ;PREPARE FLTPAT FOR LOADING INTO RMI  
2410 013726 005101 COM R1  
2411 013730 110137 177770 MOV R1,177770 ;LOAD RMI REGISTER  
2412 013734 112737 000002 177750 MOVB #HODO,CMR ;ALLOWS CACHE TAG STORE TO BE WRITTEN  
2413 ;TO CHR<15:07> DURING THE DESTINATION  
2414 ;MEMORY ACCESS OF AN INSTRUCTION ONLY  
2415 013742 012737 000015 177746 MOV #15,CCR ;NO UCB TO ENABLE TAG STORE WRITING  
2416 013750 052737 000400 177770 BIS #400,177770 ;ENABLE RMI  
2417 013756 005737 040000 TST 40000  
2418 013762 005737 100000 TST 100000 ;SELECT PAGE 4 AND READ FLOATING ADDRESS  
2419 ;SPECIFIED BY KPAR4. RMI WILL RESPOND

```

2420                                     ;RESULTING IN THE TAG STORE BEING LOADED
2421                                     ;FROM CA<21:13>
2422 013766 005737 100000             TST      100000
2423 013772 013737 177752 047322     MOV      CHR,CHR157
2424 014000 042737 000400 177770     BIC      #400,177770
2425 014006 000240                 25$:   NOP
      014010 000240                 NOP
2426 014012 005037 177572             CLR      SRO
2427 014016 005037 172516             CLR      SR3
2428 014022 012737 001015 177746     MOV      #OFF,CCR
2429 014030 105037 177750             CLRB     CMR
2430 014034 042737 000177 047322     BIC      #177,CHR157
2431 014042 023737 172350 047322     CMP      KPAR4,CHR157
2432 014050 001435                 BEQ      9$
2433 014052 012737 000006 000004     MOV      #6,4
2434 014060 005037 000006             CLR      6
2435 014064 013737 172350 047324     MOV      KPAR4,CA2113
2436 014072 012737 000007 002062     MOV      #7,LOOP
2437 014100 006237 047322             6$:   ASR      CHR157
2438 014104 006237 047324             ASR      CA2113
2439 014110 042737 100000 047322     BIC      #100000,CHR157
2440 014116 042737 100000 047324     BIC      #100000,CA2113
2441 014124 005337 002062             DEC      LOOP
2442 014130 001363                 BNE     6$
2443 014132 104406                 ERROR
                                     ;ERROR
                                     ;-----
      014134 014132                 .WORD   -2
2444                                     ;TAG STORE TESTS USING RMI REGISTER
2445                                     ;READING CHR<15:07> FOR TAG DATA (TAGD <21:13>)
2446                                     ;DID NOT RESULT IN CORRECT ADDRESS PATTERN
2447                                     ;LOADED FROM CA<21:13>.
2448 014136 047322             CHR157
2449 014140 047324             CA2113
2450 014142 000000             .WORD   0
2451 014144 006337 172350             9$:   ASL      KPAR4
2452 014150 006337 047344             ASL      FLTPAT
2453 014154 103245             BCC     1$
2454                                     ;CONTINUE TEST. ADDRESS 10000000
2455 014156 012737 000006 000004     MOV      #6,4
2456 014164 005037 000006             CLR      6
2457 014170 000464             BR      10$
2458 014172 022626             3$:   CMP      (SP)+,(SP)+
2459 014174 012737 000006 000004     4$:   MOV      #6,4
2460 014202 005037 000006             CLR      6
2461 014206 005737 001474             TST      $PASS
2462 014212 001053             BNE     10$
2463 014214 023737 000042 000046     CMP      42,46
2464 014222 001447             BEQ     10$
2465 014224 104401 014232             TYPE    ,65$
      014230 000427             BR      64$
      014232      200      122      115 65$: .ASCIZ <CRLF>/RMI REGISTER (G5179) NOT USED-SKIP HI ORDER/
                                     .EVEN
      014310             64$:
2466 014310 104401 014316             TYPE    ,67$
      014314 000412             BR      66$
      014316      040      102      111 67$: .ASCIZ / BIT ADDRESS TEST/<CRLF>
                                     .EVEN
    
```



2467	014342	000240	668:	NOP		;END OF TEST
	014342	005237	108:	INC	\$TESTN	;INCREMENT TEST COUNTER
	014344	001472				

TEST # 104 - VERIFY TAG STORE ADDRESS LINES (CA(12:1))  
2476

SBTTL TEST # 104 - VERIFY TAG STORE ADDRESS LINES (CA(12:1))  
\*\*\*\*\*  
\*TEST 104 - VERIFY TAG STORE ADDRESS LINES (CA(12:1))  
\*VERIFY TAG STORE ADDRESS LINES (CA(12:1))  
\*PROCEDURE: WRITE 0 INTO TAGG PARITY STORE ADDRESS LOCATION 0000.  
\*WRITE BIT PATTERN 000000011 INTO TAG PARITY STORE LOCATION 0001.  
\*READ TAG PARITY ADDRESS LOCATION 0000 FOR 0'S REPEAT THE ABOVE  
\*SEQUENCE, EACH TIME CHANGING THE ADDRESS LOCATION THE BIT PATTERN  
\*IS WRITTEN TO BY SHIFTING THE 1 ONE PLACE TO THE LEFT.  
\*\*\*\*\*

2477 014350 000004  
014352 014362  
014354 070032  
014356 000000  
014360 070072  
014362 012737 001015 177746 40\$:  
014370 004437 002370  
014374 014556

TST104:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION  
;DISABLE CACHE  
;LOCATE TEST CODE TO HIGH CACHE SPACE  
;ADDRESS OF START OF NEXT TEST  
;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

2478 014376 012737 000002 047344  
2479 014404 012702 040000 2\$:  
2480 014410 012703 060000  
2481 014414 063702 047344  
2482 014420 063703 047344  
2483 014424 012713 177777  
2484  
2485 014430 112737 000002 177750 1\$:  
2486  
2487  
2488 014436 012737 000015 177746  
2489 014444 005737 040000  
2490 014450 005737 000000  
2491  
2492 014454 005712  
2493 014456 005713  
2494  
2495  
2496 014460 005737 060000  
2497 014464 013701 177752  
2498 014470 000240 25\$:  
014472 000240  
2499 014474 105037 177750  
2500 014500 012737 001015 177746  
2501 014506 042701 000177  
2502 014512 005701  
2503 014514 001411  
2504 014516 013737 047344 047334  
2505 014524 006237 047334  
2506 014530 104406

MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN  
MOV #40000,R2  
MOV #60000,R3  
ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT  
ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT  
MOV #-1,(R3) ;ALL 1'S TO MAIN MEM. LOCATION  
;SPECIFIED BY R3  
MOV #HODO,CMR ;ALLOWS TAG STORE BIT TO BE  
;WRITTEN TO CHR<15:07> ONLY DURING THE  
;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.  
;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
MOV #15,CCR  
TST 40000  
TST 0 ;READ UPDATE; WRITE ALL 0'S INTO CACHE  
;TAG STORE LOCATION 0000.  
TST (R2)  
TST (R3) ;READ UPDATE;WRITE BIT PATTERN 000000011  
;INTO TAG STORE LOCATION SPECIFIED  
;BY R3'S BITS 1 THRU 12: CA<12:1>  
;LOAD TAG STORE LOCATION 0000 INTO CHR  
;SAVE CHR CONTENTS  
;INSTRUCTION 'JMP 1\$' PLACED HERE  
;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINT. MODE  
MOV #OFF,CCR ;DISABLE CACHE  
BIC #177,R1 ;INTERESTED IN ONLY BITS 15:07  
TST R1 ;CHECK FOR ALL 0'S  
BEQ 9\$ ;PASS  
MOV FLTPAT,CA121 ;SAVE CA<12:1> USED  
ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
ERROR  
-----  
;TAG STORE ADRESS LINE TESTS

2507 014532 014530

.WORD -2

```

2508                               ;READING CHR<15:07> FOR CACHE TAG STORE
2509                               ;DID NOT RESULT IN ALL 0'S
2510 014534 047334                CA121   ;PRINT CACHE TAG STORE ADDRESS LOCATION
2511                               ;USED: CA<12:1>. NOTE THAT THE 1 IN
2512                               ;THIS PATTERN WILL POINT TO THE ADDRESS
2513                               ;LINE THAT POSSIBLY CAUSES ERROR.
2514 014536 000000                .WORD   0
2515 014540 006337 047344        9$:    ASL   FLTPAT
2516 014544 022737 020000 047344    CMP   #20000,FLTPAT
2517 014552 001314                BNE   2$
2518 014554 000240                10$:   NOP
      014556 005237 001472        INC   $TESTN
  
```

```

;READING CHR<15:07> FOR CACHE TAG STORE
;DID NOT RESULT IN ALL 0'S
;PRINT CACHE TAG STORE ADDRESS LOCATION
;USED: CA<12:1>. NOTE THAT THE 1 IN
;THIS PATTERN WILL POINT TO THE ADDRESS
;LINE THAT POSSIBLY CAUSES ERROR.
;NEXT PATTERN
;HAS CACHE TAG STORE LOCAT. 4000 BEEN DONE?
;NO
;END OF TEST
;INCREMENT TEST COUNTER
  
```

2523

```
.SBTTL TEST # 105 - WRITE ALL 0'S INTO DATA STORE LOCATION 0000
*****
*TEST 105 - WRITE ALL 0'S INTO DATA STORE LOCATION 0000
*   WRITE ALL 0'S INTO DATA STORE LOCATION 0000.
*   READ ALL 0'S FROM CACHE DATA REGISTER.
*****
```

2524 014562 000004  
 014564 014574  
 014566 070004  
 014570 000000  
 014572 070042  
 014574 012737 001015 177746  
 014602 004437 002370  
 014606 014710

```
TST105:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
                          ;TEST START LOCATION
      .WORD 40$            ;LOOP ON ERROR START LOCATION
      .WORD 1$-40$+67764  ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0              ;LOOP ON ERROR END LOCATION
      .WORD 25$-40$+67764 ;DISABLE CACHE
      MOV #OFF,CCR        ;LOCATE TEST CODE TO HIGH CACHE SPACE
      JSR R4,RELCTH      ;ADDRESS OF START OF NEXT TEST
      .WORD 10$+2
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

2525 014610 005037 060000  
 2526 014614 112737 000002 177750  
 2527  
 2528  
 2529 014622 012737 000015 177746  
 2530 014630 005737 040000  
 2531 014634 005737 060000  
 2532  
 2533  
 2534 014640 005737 060000  
 2535  
 2536 014644 013737 177754 047330  
 2537 014652 000240  
 014654 000240  
 2538 014656 105037 177750  
 2539 014662 012737 001015 177746  
 2540 014670 005737 047330  
 2541 014674 001404  
 2542 014676 104406  
 014700 014676  
 2543  
 2544  
 2545  
 2546 014702 047330  
 2547 014704 000000  
 2548 014706 000240  
 014710 005237 001472

```
      CLR 60000            ;0'S TO MAIN MEMORY LOCATION
      MOV #HODO,CMR       ;ALLOWS CACHE DATA STORE BITS TO BE
                          ;WRITTEN TO CDR<15:0> ONLY DURING THE
                          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
                          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
      MOV #15,CCR        ;
      TST 40000          ;
      TST 60000          ;WRITE ALL 0'S TO DATA STORE
                          ;LOCATION 0000 FROM MAIN MEMORY
                          ;LOC. 60000
      TST 60000          ;WRITE DATA STORE BITS FROM
                          ;LOC. 0000 INTO CDR<15:0>.
      MOV CDR,CDR150     ;SAVE CDR CONTENTS
      NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP                ;FOR LOOP ON ERROR
      CLRB CMR           ;DISABLE MAINTENANCE
      MOV #OFF,CCR      ;DISABLE CACHE
      TST CDR150        ;CHECK FOR 0
      BEQ 10$           ;PASS
      ERROR             ;ERROR
      -----
      .WORD -2          ;
                          ; DATA STORE TESTS
                          ;READING CDR<15:0> DID NOT RESULT
                          ;IN ALL 0'S
                          ;PRINT CDR<15:0> DATA READ.
      CDR150            ;
      .WORD 0           ;
      NOP                ;END OF TEST
      INC $TESTN        ;INCREMENT TEST COUNTER
```

2553

```
.SBTTL TEST # 106 - WRITE ALL 1'S INTO DATA STORE LOCATION 0000
:*****
:*TEST 106 - WRITE ALL 1'S INTO DATA STORE LOCATION 0000
:*  WRITE ALL 1'S INTO DATA STORE LOCATION 0000.
:*  READ ALL 1'S FROM CACHE DATA REGISTER.
:*****
```

2554 014714 000004

```
TST106:
      SCPCNF                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
      .WORD 40$              ;TEST START LOCATION
      .WORD 1$-40$+67764    ;LOOP ON ERROR START LOCATION
      .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$-40$+67764   ;LOOP ON ERROR END LOCATION
      001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
      002370                ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2           ;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

2555 014742 012737 177777 060000

```
1$: MOV #-1,60000 ;1'S TO MAIN MEMORY LOCATION
     MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE
                       ;WRITTEN TO CDR<15:07> ONLY DURING THE
                       ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
                       ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
```

2556 014750 112737 000002 177750

2557  
2558

2559 014756 012737 000015 177746

```
MOV #15,CCR
TST 40000
TST 60000 ;WRITE ALL 1'S TO DATA STORE
           ;LOCATION 0000 FROM MAIN MEMORY
           ;LOC. 60000
           ;WRITE DATA STORE BITS FROM
           ;LOC. 0000 INTO CDR<15:07>.
           ;SAVE CDR CONTENTS
           ;INSTRUCTION 'JMP 1$' PLACED HERE
           ;FOR LOOP ON ERROR
           ;DISABLE MAINTENANCE
           ;DISABLE CACHE
           ;CHECK ALL 1'S
           ;PASS
           ;ERROR
           ;-----
```

2560 014764 005737 040000

2561 014770 005737 060000

2562  
2563

2564 014774 005737 060000

2565  
2566 015000 013737 177754 047330

```
25$: MOV CDR,CDR150 ;DATA STORE TESTS
      NOP ;READING CDR<15:0> DID NOT RESULT
      NOP ;IN ALL 1'S
      CLRB CMR ;PRINT CDR<15:0> DATA READ.
      MOV #OFF,CCR ;DISABLE CACHE
      CMP #-1,CDR150 ;CHECK ALL 1'S
      BEQ 10$ ;PASS
      ERROR ;ERROR
      -----
```

2567 015006 000240

2568 015010 000240

2569 015012 105037 177750

2570 015016 012737 001015 177746

2571 015024 022737 177777 047330

2572 015032 001404

2573 015034 104406

2574  
2575

2576 015036 015034

2577 015040 047330

2578 015042 000000

```
10$: .WORD 0 ;END OF TEST
      NOP ;INCREMENT TEST COUNTER
      INC $TESTN
```

2579 015044 000240

2580 015046 005237 001472

2583

```
.SBTTL TEST # 107 - WRITE FLOATING 1 PATRN INTO DATA STORE LOC 0
*****
*TEST 107 - WRITE FLOATING 1 PATRN INTO DATA STORE LOC 0
*   WRITE FLOATING 1 PATTERN INTO DATA STORE LOCATION 0000.
*   READ FLOATING 1 PATTERN FROM CACHE DATA REGISTER.
*****
```

015052  
 2584 015052 000004  
 015054 015064  
 015056 070014  
 015060 000000  
 015062 070052  
 015064 012737 001015 177746  
 015072 004437 002370  
 015076 015232

```
TST107:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;DISABLE CACHE
;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

2585 015100 012737 000001 047344  
 2586 015106 013737 047344 060000 2\$:  
 2587 015114 112737 000002 177750 1\$:  
 2588  
 2589  
 2590 015122 012737 000015 177746  
 2591 015130 005737 040000  
 2592 015134 005737 060000  
 2593  
 2594  
 2595 015140 005737 060000  
 2596  
 2597 015144 013737 177754 047330  
 2598 015152 000240 25\$:  
 015154 000240  
 2599 015156 105037 177750  
 2600 015162 012737 001015 177746  
 2601 015170 023737 047344 047330  
 2602 015176 001410  
 2603 015200 013737 047344 047332  
 2604 015206 104406

```
MOV #1,FLTPAT ;1ST FLOATING 1 PATTERN: 000001
MOV FLTPAT,60000 ;FLOATING PATTERN TO MAIN MEMORY
MOV#B #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE
;WRITTEN TO CDR<15:07> ONLY DURING THE
;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
;NO UCB SO AS TO WRITE ENABLE CACHE STORE
MOV #15,CCR
TST 40000
TST 60000 ;WRITE FLOATING 1 PATTERN TO DATA STORE
;LOCATION 0000 FROM MAIN MEMORY
;LOC. 60000
TST 60000 ;WRITE DATA STORE BITS FROM
;LOC. 0000 INTO CDR<15:0>.
MOV CDR,CDR150 ;SAVE CDR CONTENTS
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINTENANCE
MOV #OFF,CCR ;DISABLE CACHE
CMP FLTPAT,CDR150 ;CHECK FOR CORRECT PATTERN
BEQ 9$ ;PASS
MOV FLTPAT,EXDAT6 ;SAVE FLOATING PATTERN FOR TYPEOUT
ERROR ;ERROR
-----
```

015210 015206  
 2605  
 2606  
 2607  
 2608 015212 047332  
 2609 015214 047330  
 2610 015216 000000  
 2611 015220 006337 047344 9\$:  
 2612 015224 103401  
 2613 015226 000727  
 2614 015230 000240 10\$:  
 015232 005237 001472

```
.WORD .-2 ; DATA STORE TESTS
;READING CDR<15:0> DID NOT RESULT
;IN CORRECT FLOATING 1 PATTERN
;PRINT FLOATING PATTERN EXPECTED
;PRINT CDR<15:0> DATA READ.
EXDAT6
CDR150
WORD 0
ASL FLTPAT ;NEXT PATTERN
BCS 10$ ;IF PATTERN 100000 HAS BEEN DONE;FINISHED
BR 2$ ;IF NOT, NEXT PASS
NOP ;END OF TEST
INC $TESTN ;INCREMENT TEST COUNTER
```

TEST # 110 - CLEAR ALL LOW CACHE DATA STORE LOCATIONS  
2619

.SBTTL TEST # 110 - CLEAR ALL LOW CACHE DATA STORE LOCATIONS  
\*\*\*\*\*  
:TEST 110 - CLEAR ALL LOW CACHE DATA STORE LOCATIONS  
:\* WRITE ALL 0'S INTO ALL LOW CACHE DATA STORE LOCATIONS (0000 TO 3777).  
:\* READ ALL 0'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.  
\*\*\*\*\*

TST110:  
2620 015236 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
015240 015250 .WORD 40\$ ;TEST START LOCATION  
015242 070024 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
015244 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
015246 070054 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
015250 012737 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE  
015256 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
015262 015430 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

2621 015264 012705 060000 MOV #60000,R5 ;ADDRESS 60000 INTO R5  
2622 015270 005025 5\$: CLR (R5)+ ;CLEAR MAIN MEMORY LOW CACHE AREA  
2623 015272 020527 070000 CMP R5,#70000 ;FINISHED?  
2624 015276 001374 BNE 5\$ ;NO  
2625 015300 012705 060000 MOV #60000,R5 ;START WITH ADDRESS 60000  
2626 015304 012703 040000 MOV #40000,R3 ;ADDRESS 40000 INTO R3  
2627 015310 112737 000002 177750 1\$: MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE  
2628 ;WRITTEN TO CDR<15:0> ONLY DURING THE  
2629 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.  
2630 015316 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
2631 015324 005713 TST (R3)  
2632 015326 005715 TST (R5)  
2633 015330 005715 TST (R5) ;WRITE ALL 0'S TO DATA STORE FROM MAIN MEM.  
2634 015332 013737 177754 047330 MOV CDR,CDR150 ;WRITE DATA STORE BITS INTO CDR<15:0>  
2635 015340 000240 25\$: NOP ;SAVE CDR CONTENTS  
015342 000240 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
2636 015344 105037 177750 CLRB CMR ;FOR LOOP ON ERROR  
2637 015350 012737 001015 177746 MOV #OFF,CCR ;DISABLE MAINTENANCE  
2638 015356 022737 000000 047330 CMP #0,CDR150 ;DISABLE CACHE  
2639 015364 001411 BEQ 9\$ ;CHECK ALL 0'S  
2640 015366 010537 047314 MOV R5,CA210+2 ;PASS  
2641 015372 005037 047312 CLR CA210 ;SAVE ADDRESS USED THIS PASS  
2642 015376 104406 ERROR ;ERROR  
;-----

015400 015376 .WORD -2  
2643 ; DATA STORE TESTS  
2644 ;READING CDR<15:0> DID NOT RESULT  
2645 ;IN ALL 0'S  
2646 015402 047312 CA210 ;PRINT CA<21:0> USED  
2647 015404 047330 CDR150 ;PRINT CDR<15:0> DATA READ.  
2648 015406 000000 .WORD 0  
2649 015410 062705 000002 9\$: ADD #2,R5 ;NEXT CACHE LOCATION  
2650 015414 062703 000002 ADD #2,R3  
2651 015420 022705 070000 CMP #70000,R5 ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?  
2652 015424 001331 BNE 1\$ ;NO  
2653 015426 000240 10\$: NOP ;END OF TEST  
015430 005237 001472 INC \$TESTN ;INCREMENT TEST COUNTER

.SBTTL TEST # 111 - SET ALL LOW CACHE DATA STORE LOCATIONS  
 :\*\*\*\*\*  
 :TEST 111 - SET ALL LOW CACHE DATA STORE LOCATIONS  
 :\* WRITE ALL 1'S INTO ALL LOW CACHE DATA STORE LOCATIONS (0000 TO 3777).  
 :\* READ ALL 1'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.  
 :\*\*\*\*\*

TST111:  
 2659 015434 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 015436 015446 .WORD 40\$ ;TEST START LOCATION  
 015440 070026 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
 015442 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 015444 070056 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
 015446 012737 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE  
 015454 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
 015460 015630 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

2660 015462 012705 060000 MOV #60000,R5 ;ADDRESS 60000 INTO R5  
 2661 015466 012725 177777 5\$: MOV #-1,(R5)+ ;1'S TO MAIN MEMORY LOW CACHE AREA  
 2662 015472 020527 070000 CMP R5,#70000 ;FINISHED?  
 2663 015476 001373 BNE 5\$ ;NO  
 2664 015500 012705 060000 MOV #60000,R5 ;START WITH ADDRESS 60000  
 2665 015504 012703 040000 MOV #40000,R3 ;ADDRESS 40000 INTO R3  
 2666 015510 112737 000002 177750 1\$: MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE  
 2667 ;WRITTEN TO CDR<15:0> ONLY DURING THE  
 2668 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.  
 2669 015516 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
 2670 015524 005713 TST (R3)  
 2671 015526 005715 TST (R5)  
 2672 015530 005715 TST (R5) ;WRITE ALL 1'S TO DATA STORE FROM MAIN MEM.  
 2673 015532 013737 177754 047330 MOV CDR,CDR150 ;WRITE DATA STORE BITS INTO CDR<15:0>  
 2674 015540 000240 25\$: NOP ;SAVE CDR CONTENTS  
 015542 000240 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 2675 015544 105037 177750 CLR B CMR ;FOR LOOP ON ERROR  
 2676 015550 012737 001015 177746 MOV #OFF,CCR ;DISABLE MAINTENANCE  
 2677 015556 022737 177777 047330 CMP #-1,CDR150 ;DISABLE CACHE  
 2678 015564 001411 BEQ 9\$ ;CHECK ALL 1'S  
 2679 015566 010537 047314 MOV R5,CA210+2 ;PASS  
 2680 015572 005037 047312 CLR CA210 ;SAVE ADDRESS USED THIS PASS  
 2681 015576 104406 ERROR ;ERROR  
 ;-----  
 015600 015576 .WORD .-2  
 2682 ; DATA STORE TESTS  
 2683 ;READING CDR<15:0> DID NOT RESULT  
 2684 ;IN ALL 1'S  
 2685 015602 047312 CA210 ;PRINT CA<21:0> USED  
 2686 015604 047330 CDR150 ;PRINT CDR<15:0> DATA READ.  
 2687 015606 000000 .WORD 0  
 2688 015610 062705 000002 9\$: ADD #2,R5 ;NEXT CACHE LOCATION  
 2689 015614 062703 000002 ADD #2,R3  
 2690 015620 022705 070000 CMP #70000,R5 ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?  
 2691 015624 001331 BNE 1\$ ;NO  
 2692 015626 000240 10\$: NOP ;END OF TEST  
 015630 005237 001472 INC \$TESTN ;INCREMENT TEST COUNTER



2697

.SBTTL TEST # 112 - CLEAR ALL HIGH CACHE DATA STORE LOCATIONS  
\*\*\*\*\*  
\*TEST 112 - CLEAR ALL HIGH CACHE DATA STORE LOCATIONS  
\* WRITE ALL 0'S INTO ALL HIGH CACHE DATA STORE LOCATIONS (4000 TO 7777).  
\* READ ALL 0'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.  
\*\*\*\*\*  
TST112:

2698 015634 000004

015636 015646  
015640 060024  
015642 000000  
015644 060054  
015646 012737  
015654 004437  
015660 016026

001015 177746  
002342

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
.WORD 40\$ ;LOOP ON ERROR START LOCATION  
.WORD 1\$-40\$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 0 ;LOOP ON ERROR END LOCATION  
.WORD 25\$-40\$+57764 ;DISABLE CACHE  
MOV #OFF,CCR ;LOCATE TEST CODE TO LOW CACHE SPACE  
JSR R4,RELCTL ;ADDRESS OF START OF NEXT TEST  
.WORD 10\$+2

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO LOW CACHE SPACE

2699 015662 012705 070000  
2700 015666 005025  
2701 015670 020527 100000  
2702 015674 001374  
2703 015676 012705 070000  
2704 015702 012703 050000  
2705 015706 112737 000002 177750  
2706  
2707  
2708 015714 012737 000015 177746  
2709 015722 005713  
2710 015724 005715  
2711 015726 005715  
2712 015730 013737 177754 047330  
2713 015736 000240  
015740 000240  
2714 015742 105037 177750  
2715 015746 012737 001015 177746  
2716 015754 022737 000000 047330  
2717 015762 001411  
2718 015764 010537 047314  
2719 015770 005037 047312  
2720 015774 104406

5\$: MOV #70000,R5 ;ADDRESS 70000 INTO R5  
CLR (R5)+ ;CLEAR MAIN MEMORY HI CACHE AREA  
CMP R5,#100000 ;FINISHED?  
BNE 5\$ ;NO  
MOV #70000,R5 ;START WITH ADDRESS 70000  
MOV #50000,R3 ;ADDRESS 50000 INTO R3  
1\$: MOVB #H000,CMR ;ALLOWS CACHE DATA STORE BITS TO BE  
;WRITTEN TO CDR<15:0> ONLY DURING THE  
;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.  
;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
MOV #15,CCR  
TST (R3)  
TST (R5) ;WRITE ALL 0'S TO DATA STORE FROM MAIN MEM.  
TST (R5) ;WRITE DATA STORE BITS INTO CDR<15:0>  
25\$: MOV CDR,CDR150 ;SAVE CDR CONTENTS  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINTENANCE  
MOV #OFF,CCR ;DISABLE CACHE  
CMP #0,CDR150 ;CHECK ALL 0'S  
BEQ 9\$ ;PASS  
MOV R5,CA210+2 ;SAVE ADDRESS USED THIS PASS  
CLR CA210  
ERROR ;ERROR  
;-----

015776 015774

2721  
2722  
2723  
2724 016000 047312  
2725 016002 047330  
2726 016004 000000  
2727 016006 062705 000002  
2728 016012 062703 000002  
2729 016016 022705 100000  
2730 016022 001331  
2731 016024 000240  
016026 005237 001472

.WORD -2  
; DATA STORE TESTS  
;READING CDR<15:0> DID NOT RESULT  
;IN ALL 0'S  
;PRINT CA<21:0> USED  
;PRINT CDR<15:0> DATA READ.  
9\$: .WORD 0 ;NEXT CACHE LOCATION  
ADD #2,R5  
ADD #2,R3  
CMP #100000,R5 ;HAVE ALL HIGH CACHE LOCATIONS BEEN DONE?  
BNE 1\$ ;NO  
10\$: NOP ;END OF TEST  
INC \$TESTN ;INCREMENT TEST COUNTER

DATA STORE LOCATIONS

```

.SBTTL TEST # 113 - SET ALL HIGH CACHE DATA STORE LOCATIONS
*****
*TEST 113 - SET ALL HIGH CACHE DATA STORE LOCATIONS
*   WRITE ALL 1'S INTO ALL HIGH CACHE DATA STORE LOCATIONS (4000 TO 7777).
*   READ ALL 1'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.
*****

```

```

2737 016032 000004
      016034 016044
      016036 060026
      016040 000000
      016042 060056
      016044 012737 001015 177746
      016052 004437 002342
      016056 016226

```

```

TST113:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
                          ;TEST START LOCATION
      .WORD 40$             ;LOOP ON ERROR START LOCATION
      .WORD 1$-40$+57764   ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0               ;LOOP ON ERROR END LOCATION
      .WORD 25$-40$+57764 ;DISABLE CACHE
      MOV #OFF,CCR          ;LOCATE TEST CODE TO LOW CACHE SPACE
      JSR R4,RELCTL        ;ADDRESS OF START OF NEXT TEST
      .WORD 10$+2

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE

```

```

2738 016060 012705 070000
2739 016064 012725 177777
2740 016070 020527 100000
2741 016074 001373
2742 016076 012705 070000
2743 016102 012703 050000
2744 016106 112737 000002 177750
2745
2746
2747 016114 012737 000015 177746
2748 016122 005713
2749 016124 005715
2750 016126 005715
2751 016130 013737 177754 047330
2752 016136 000240
      016140 000240
2753 016142 105037 177750
2754 016146 012737 001015 177746
2755 016154 022737 177777 047330
2756 016162 001411
2757 016164 010537 047314
2758 016170 005037 047312
2759 016174 104406

```

```

      5$: MOV #70000,R5      ;ADDRESS 70000 INTO R5
          MOV #-1,(R5)+     ;1'S TO MAIN MEMORY HI CACHE AREA
          CMP R5,#100000    ;FINISHED?
          BNE 5$            ;NO
          MOV #70000,R5     ;START WITH ADDRESS 70000
          MOV #50000,R3     ;ADDRESS 50000 INTO R3
      1$: MOVB #HODO,CMR    ;ALLOWS CACHE DATA STORE BITS TO BE
                          ;WRITTEN TO CDR<15:0> ONLY DURING THE
                          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
                          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
          MOV #15,CCR
          TST (R3)
          TST (R5)
          TST (R5)
          MOV CDR,CDR150   ;WRITE ALL 1'S TO DATA STORE FROM MAIN MEM.
                          ;WRITE DATA STORE BITS INTO CDR<15:0>
      25$: NOP              ;SAVE CDR CONTENTS
          NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
          CLRB CMR         ;FOR LOOP ON ERROR
                          ;DISABLE MAINTENANCE
          MOV #OFF,CCR     ;DISABLE CACHE
          CMP #-1,CDR150  ;CHECK ALL 1'S
          BEQ 9$           ;PASS
          MOV R5,CA210+2  ;SAVE ADDRESS USED THIS PASS
          CLR CA210
          ERROR
          -----

```

```

      016176 016174
2760
2761
2762
2763 016200 047312
2764 016202 047330
2765 016204 000000
2766 016206 062705 000002
2767 016212 062703 000002
2768 016216 022705 100000
2769 016222 001331
2770 016224 000240
      016226 005237 001472

```

```

      .WORD -2
      CA210
      CDR150
      .WORD 0
      9$: ADD #2,R5         ;NEXT CACHE LOCATION
          ADD #2,R3
          CMP #100000,R5  ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
          BNE 1$
      10$: NOP
          INC $TESTN      ;END OF TEST
                          ;INCREMENT TEST COUNTER

```

SBTTL TEST # 114 - VERIFY CACHE DATA STORE ADDRESS LINES (CA(12:1))  
\*\*\*\*\*  
\*TEST 114 - VERIFY CACHE DATA STORE ADDRESS LINES (CA(12:1))  
\* VERIFY CACHE DATA STORE ADDRESS LINES (CA(12:1))  
\*\*\*\*\*  
TST114:

2775 016232 000004  
016234 016244  
016236 070040  
016240 000000  
016242 070100  
016244 012737 001015 177746 40\$:  
016252 004437 002370  
016256 016450

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
.WORD 40\$ ;LOOP ON ERROR START LOCATION  
.WORD 1\$-40\$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 0 ;LOOP ON ERROR END LOCATION  
.WORD 25\$-40\$+67764 ;DISABLE CACHE  
MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE  
JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST  
.WORD 10\$+2

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

2776 016260 000240  
2777  
2778  
2779 016262 005037 060000  
2780 016266 012737 000002 047344  
2781 016274 012702 040000 2\$:  
2782 016300 012703 060000  
2783 016304 063702 047344  
2784 016310 063703 047344  
2785 016314 012713 177777  
2786  
2787 016320 112737 000002 177750 1\$:  
2788  
2789  
2790 016326 012737 000015 177746  
2791 016334 005737 040000  
2792 016340 005737 060000  
2793  
2794 016344 005712  
2795 016346 005713  
2796  
2797  
2798 016350 005737 060000  
2799  
2800 016354 013701 177754  
2801 016360 000240 25\$:  
016362 000240  
2802 016364 105037 177750  
2803 016370 012737 001015 177746  
2804 016376 005701  
2805 016400 001411  
2806  
2807 016402 013737 047344 047334  
2808 016410 006237 047334  
2809 016414 104406

NOP ;WHEN THE TEST IS RELOCATED,ADDRESS 70000  
;WILL BE LOADED WITH ALL 1'S DURING TEST  
CLR 60000 ;ALL 0'S TO MAIN MEMORY  
MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN  
MOV #40000,R2  
MOV #60000,R3  
ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT  
ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT  
MOV #-1,(R3) ;ALL 1'S TO MAIN MEM. LOCATION  
;SPECIFIED BY R3  
MOV #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE  
;WRITTEN TO CDR<15:0> ONLY DURING THE  
;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.  
;NO UCB SO AS TO WRITE ENABLE CACHE STORE  
MOV #15,CCR  
TST 40000  
TST 60000 ;WRITE ALL 0'S FROM MAIN MEM. LOCATION  
;60000 INTO CACHE DATA STORE LOCATION 0000.  
TST (R2)  
TST (R3) ;WRITE ALL 1'S FROM MAIN MEM. INTO  
;CACHE DATA STORE LOCAT. SPECIFIED  
;BY R3'S BITS 1 THRU 12: CA<12:1>  
TST 60000 ;LOAD DATA FROM CACHE DATA STORE LOCATION  
;0000 INTO CDR.  
MOV CDR,R1 ;SAVE CDR DATA  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINT. MODE  
MOV #OFF,CCR ;DISABLE CACHE  
TST R1 ;CHECK FOR ALL 0'S  
BEQ 9\$ ;PASS  
MOV FLTPAT,CA121 ;SAVE CA<12:1> USED  
ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
ERROR ;ERROR  
-----

2810 016416 016414

.WORD .-2 ;DATA STORE TESTS- ADDRESS LINE VERIFICATION

2811  
2812  
2813 016420 047334 CA121  
2814  
2815  
2816  
2817 016422 000000  
2818 016424 006337 047344 9\$: ASL FLTPAT  
2819 016430 022737 020000 047344 CMP #20000,FLTPAT  
2820 016436 001316 BNE 2\$  
2821 016440 012737 000240 070000 MOV #240,70000  
2822 016446 000240 10\$: NOP  
016450 005237 001472 INC \$TESTN

:READING CDR<15:0> FOR CACHE DATA STORE  
:DID NOT RESULT IN ALL 0'S  
:PRINT CACHE DATA STORE ADDRESS LOCATION  
:USED: CA<12:1>. NOTE THAT THE 1 IN  
:THIS PATTERN WILL POINT TO THE ADDRESS  
:LINE THAT POSSIBLY CAUSES ERROR.  
:NEXT PATTERN  
:HAS CACHE DATA STORE LOCAT. 4000 BEEN DONE?  
:NO  
:RESTORE OVERWRITTEN LOCATION 7000 WITH A NOP  
:END OF TEST  
:INCREMENT TEST COUNTER

2830

```
.SBTTL TEST # 115 - CHECK EQUAL DATA COMPARISON CONDITION
*****
*TEST 115 - CHECK EQUAL DATA COMPARISON CONDITION
*   VERIFY THAT AN EQUAL DATA COMPARISON CONDITION CAN EXIST
*   BY COMPARING TAG STORE DATA AND CA<21:13>
*   UNDER THE FOLLOWING CONDITIONS CMR<15:13> SHOULD RESULT IN ALL
*   1'S INDICATING A MATCH :
*   TAG STORE DATA AND CA<21:13> ALL 0'S
*****
```

```
2831 016454 000004
      016456 016466
      016460 070000
      016462 000000
      016464 070036
      016466 012737 001015 177746 40$:
      016474 004437 002370
      016500 016644
```

```
TST115:
SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                ;ERROR/LOOP ON TEST
                ;TEST START LOCATION
                ;LOOP ON ERROR START LOCATION
                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                ;LOOP ON ERROR END LOCATION
                ;DISABLE CACHE
                ;LOCATE TEST CODE TO HIGH CACHE SPACE
                ;ADDRESS OF START OF NEXT TEST

                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                ;ARE RELOCATED TO HI CACHE SPACE
```

```
2832 016502 112737 000002 177750 1$:
2833
2834
2835
2836 016510 012737 000015 177746
2837 016516 005737 040000
2838 016522 005737 000000
2839
2840 016526 005737 000000
2841
2842
2843 016532 013737 177750 047340
2844 016540 000240
      016542 000240
2845 016544 105037 177750
2846 016550 012737 001015 177746
2847 016556 042737 017777 047340
2848 016564 022737 160000 047340
2849 016572 001423
2850 016574 012737 000007 047336
2851 016602 012737 000015 002062
2852 016610 006237 047340
2853 016614 042737 100000 047340
2854 016622 005337 002062
2855 016626 001370
2856 016630 104406
```

```
MOVB #HODO,CMR ;ALLOWS COMPARED RESULTS TO BE
                ;WRITTEN TO CMR<15:13> ONLY DURING
                ;THE DESTINATION MEMORY ACCESS
                ;OF AN INSTRUCTION
                ;NO UCB SO AS TO WRITE ENABLE CACHE STORE

MOV #15,CCR
TST 4000
TST 0 ;WRITE ALL 0'S INTO TAG STORE LOCATION 0000
      ;FROM CACHE ADDRESS CA<21:13>
      ;PLACE ALL 0'S ON CA<21:13> FOR COMPARISON
      ;WITH TAG STORE DATA. WRITE COMPARED
      ;RESULTS INDICATION IN CMR<15:13>
      ;SAVE CMR DATA
      ;INSTRUCTION 'JMP 1$' PLACED HERE
      ;FOR LOOP ON ERROR
      ;DISABLE MAINTENANCE MODE
      ;DISABLE CACHE

MOV CMR,CM1513
NOP
NOP
CLR B
MOV #OFF,CCR
BIC #17777,CM1513
CMP #160000,CM1513 ;CHECK THAT CMR<15:13> ALL 1'S
BEQ 10$ ;PASS
MOV #7,EXDAT1 ;INDICATE EXPECTED CMR<15:13>
MOV #13,LOOP ;ERROR;PREPARE CM1513. FOR TYPEOUT
ASR CM1513
BIC #100000,CM1513
DEC LOOP
BNE 2$
ERROR ;ERROR
      ;-----
```

```
      016632 016630
2857
2858
2859
2860 016634 047336
2861 016636 047340
2862 016640 000000
```

```
.WORD -2
EXDAT1
CM1513
WORD 0
;COMPARE TAG STORE & CA<21:13> TESTS
;BITS 15 THRU 13 OF CMR DID NOT READ
;AS ALL 1'S
;PRINT CMR<15:13> DATA EXPECTED
;PRINT CMR<15:13> DATA RECEIVED
```

2863 016642 000240  
016644 005237 001472

108:    NOP  
          INC      \$TESTN

:END OF TEST  
:INCREMENT TEST COUNTER

TEST # 116 - UNEQUAL DATA COMPARISON CONDITION CAN BE DETECTED

2871

```

.SBTTL TEST # 116 - UNEQUAL DATA COMPARISON CONDITION CAN BE DETECTED
*****
*TEST 116 - UNEQUAL DATA COMPARISON CONDITION CAN BE DETECTED
*  VERIFY THAT AN UNEQUAL DATA COMPARISON CONDITION CAN BE DETECTED
*  BY COMPARING TAG STORE AND CA<21:13>
*  UNDER THE FOLLOWING CONDITIONS FOR TAG STORE DATA AND CA<21:13>
*  CMR<15:13> SHOULD READ AS SPECIFIED IN TABLE DEFINED BY TAGS 30$
*  TO 38$.
*****

```

TST116:

```

016650
2872 016650 000411
2873
2874
2875 016652 000006
2876 016654 000005
2877 016656 000005
2878 016660 000005
2879 016662 000005
2880 016664 000003
2881 016666 000003
2882 016670 000003
2883 016672 000003
2884 016674 000004
016674 000004

```

```

          BR          39$          ;BRANCH OVER TABLE
          CMR<15:13>          CA<21:13>          TAG STORE
          -----          -----          -----
30$:      .WORD          6          : 001          000
31$:      .WORD          5          : 002          000
32$:      .WORD          5          : 004          000
33$:      .WORD          5          : 010          000
34$:      .WORD          5          : 020          000
35$:      .WORD          3          : 040          000
36$:      .WORD          3          : 100          000
37$:      .WORD          3          : 200          000
38$:      .WORD          3          : 400          000
39$:
          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          .WORD          40$          ;TEST START LOCATION
          .WORD          1$-40$+67764          ;LOOP ON ERROR START LOCATION
          .WORD          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD          25$-40$+67764          ;LOOP ON ERROR END LOCATION
001015 177746 40$:      MOV          #OFF,CCR          ;DISABLE CACHE
002370          JSR          R4,RELCTH          ;LOCATE TEST CODE TO HIGH CACHE SPACE
016706 012737          .WORD          10$+2          ;ADDRESS OF START OF NEXT TEST
016714 004437
016720 017236

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

```

```

2885 016722 012737 000200 172350 2$:      MOV          #200,KPAR4          ;KPAR4 CONTAINS THE FLOATING 1 PATTERN
2886          ;AND REPRESENTS THE THE PAGE ADDRESS FIELD
2887          ;DATA USED BY MEMORY MNGMNT. TO CONSTRUCT
2888          ;THE PHYSICAL ADDRESS. 200 IS THE 1ST
2889          ;FLOATING 1 PATTERN WHICH WILL BE CONSTRUCTED
2890          ;TO ADDRESS 20000.
2891 016730 012701 016652          MOV          #30$,R1          ;SAVE ADDRESS OF FIRST CMR<15:13>
2892          ;EXPECTED DATA
2893 016734 012737 070254 000004 1$:      MOV          #7$-2$+70000,4          ;ALLOW FOR NEX TRAP
2894 016742 012737 000340 000006          MOV          #340,6
2895 016750 112737 000002 177750          MOVB         #HODO,CMR          ;ALLOWS COMPARED RESULTS TO BE WRITTEN
2896          ;TO CMR<15:13> ONLY DURING THE DESTINATION
2897          ;MEMCRY ACCESS OF AN INSTRUCTION.
2898 016756 012737 000015 177746          MOV          #15,CCR          ;NO UCB SO AS TO WRITE CACHE STORE
2899 016764 012737 000001 177572          MOV          #1,SRO          ;ENABLE MEMORY MNGMNT.
2900 016772 012737 000020 172516          MOV          #20,SR3          ;ENABLE 22-BIT MAPPING
2901 017000 005737 040000          TST         40000
2902 017004 005737 000000          TST         0
2903 017010 005737 100000          TST         100000
2904
2905

```

```

;WRITE ALL 0'S TO TAG STORE LOCATION 0000
;CHOOSSES KPAR4 FOR ADDRESSING: PHYSICAL
;ADDRESS WILL BE DETERMINED BY FLOATING
;PATTERN USED IN KPAR4.

```

```

2906                                     ;PLACE DLOATING PATTERN ON CA<21:13>
2907                                     ;FOR COMPARISON WITH TAG STORE DATA
2908                                     ;AT LOCATION 0000. WRITE THE COMPARED
2909                                     ;RESULT INDICATION INTO CMR<15:13>.
2910 017014 000240                       NOP
2911 017016 000240                       NOP
2912 017020 013737 177750 047340         MOV    CMR,CM1513      ;SAVE CMR DATA
2913 017026 000240 25$:                  NOP                    ;INSTRUCTION 'JMP 1$' PLACED HERE
    017030 000240                       NOP                    ;FOR LOOP ON ERROR
2914 017032 005037 177572                 CLR    SRO            ;DISABLE MEM MNGMENT.
2915 017036 005037 172516                 CLR    SR3
2916 017042 012737 001015 177746         MOV    #OFF,CCR      ;DISABLE CACHE
2917 017050 105037 177750                 CLR    CMR           ;DISABLE MAINTENANCE
2918 017054 012737 000015 002062         MOV    #13,LOOP     ;PREPARE CM1513. FOR ERROR CHECK
2919 017062 006237 047340 5$:           ASR    CM1513
2920 017066 042737 100000 047340         BIC    #100000,CM1513
2921 017074 005337 002062                 DEC    LOOP
2922 017100 001370                         BNE    5$
2923 017102 023711 047340                 CMP    CM1513,(R1)   ;COMPARE CMR<15:13> RECEIVED WITH
2924                                     ;EXPECTED
2925 017106 001426                         BEQ    9$            ;PASS
2926 017110 013737 172350 047324         MOV    KPAR4,CA2113 ;SAVE PATTERN USED FOR CA<21:13>
2927 017116 012737 000007 002062         MOV    #7,LOOP     ;PREPARE CA2113 FOR PRINTOUT
2928 017124 006237 047324 6$:           ASR    CA2113
2929 017130 042737 100000 047324         BIC    #100000,CA2113
2930 017136 005337 002062                 DEC    LOOP
2931 017142 001370                         BNE    6$
2932 017144 011137 047336                 MOV    (R1),EXDAT1 ;PREPARE EXPECTED FOR PRINTOUT
2933 017150 104406                         ERROR                ;ERROR
    017152 017150                         .WORD  -2            ;-----
2934                                     ;COMPARE TAG STORE AND CA<21:13> TESTS
2935                                     ;CMR<15:13> DID NOT READ CORRECTLY
2936 017154 047324                         CA2113                ;PRINT CA<21:13> PATTERN USED ON ADDRESS
2937                                     ;LINES
2938 017156 047336                         EXDAT1                ;PRINT CMR<15:13> EXPECTED
2939 017160 047340                         CM1513                ;PRINT CMR<15:13> RECEIVED
2940 017162 000000                         .WORD  0              ;PRINT TERMINATE
2941 017164 006337 172350 9$:           ASL    KPAR4         ;NEXT PATTERN;IF PHYSICAL ADDRESS
2942                                     ;10000000 HAS BEEN DONE; FINISHED
2943 017170 103414                         BCS    8$
2944 017172 005721                         TST    (R1)+
2945                                     ;
2946 017174 000657                         BR     1$
2947 017176 005037 177572 7$:           CLR    SRO            ;NOT FINISHED; POINT TO NEXT
2948 017202 005037 172516                 CLR    SR3           ;CMR<15:13> EXPECTED
2949 017206 105037 177750                 CLR    CMR           ;CONTINUE WITH TEST
2950 017212 012737 001015 177746         MOV    #OFF,CCR      ;DISABLE MEM. MNGMNT.
2951 017220 022626                         CMP    (SP)+,(SP)+   ;DISABLE MAINTENANCE
2952 017222 012737 000006 000004 8$:     MOV    #6,4          ;DISABLE CACHE
2953 017230 005037 000006                 CLR    6              ;RESTORE STACK DUE TO INTERRUPT
2954 017234 000240 10$:                   NOP                    ;RESTORE VECTORS
    017236 005237 001472                 INC    $TESTN        ;END OF TEST
    ;INCREMENT TEST COUNTER
    
```



2958

.SBTTL TEST # 117 - TEST FLUSH IN PROGRESS BIT  
\*\*\*\*\*  
\*TEST 117 - TEST FLUSH IN PROGRESS BIT  
\* VERIFY FLUSH IN PROGRESS BIT WILL SET AS A RESULT OF FLUSH  
\*\*\*\*\*  
TST117:

2959 017242 000004

SCPCND

;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION

017244 017254  
017246 017254  
017250 000000  
017252 017310

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

2960 017254 005002

40\$:  
1\$:

CLR R2

;INITIALIZE COUNTER

2961 017256 005037 002070

CLR FAIL1

;INITIALIZE ERROR FLAG

2962 017262 052737 000400 177746

BIS #FC,CCR

;FLUSH CACHE

2963 017270 032737 010000 177746

BIT #VCIP,CCR

;VERIFY FLUSH IN PROGRESS

2964 017276 001002

BNE 3\$

;VCIP BIT IS SET

2965 017300 005237 002070

INC FAIL1

;INDICATE ERROR

2966 017304 005302

3\$:

DEC R2

;WAIT DELAY FOR FLUSH TO COMPLETE

2967 017306 001376

BNE 3\$

2968 017310 000240

25\$:

NOP

;INSTRUCTION 'JMP 1\$' PLACED HERE

2969 017314 005737 002070

NOP

;FOR LOOP ON ERROR

2970 017320 001403

TST FAIL1

;IS THERE ERROR

2971 017322 104406

BEQ 10\$

;PASS

017324 017322

.WORD -2

;ERROR

2972

;-----  
;FLUSH CACHE TESTS

2973

;FLUSH IN PROGRESS BIT(VCIP) FAILED

2974

;TO SET AS A RESULT OF SETTING CACHE FLUSH BIT

2975 017326 000000

.WORD 0

2976 017330 000240

10\$:

NOP

;END OF TEST

017332 005237 001472

INC \$TESTN

;INCREMENT TEST COUNTER

2980

.SBTTL TEST # 120 - TEST FLUSH IN PROGRESS BIT(VCIP) WILL RESET  
\*\*\*\*\*  
\*TEST 120 - TEST FLUSH IN PROGRESS BIT(VCIP) WILL RESET  
\* VERIFY FLUSH IN PROGRESS BIT(VCIP) WILL RESET ON COMPLETION OF FLUSH  
\*\*\*\*\*  
TST120:

2981 017336 000004

SCPCND

;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION

017340 017350  
017342 017350  
017344 000000  
017346 017374  
017350

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

2982 017350 005002  
2983 017352 052737 000400 177746  
2984 017360 032737 010000 177746  
2985 017366 001407  
2986 017370 005302  
2987 017372 001372  
2988 017374 000240  
017376 000240  
2989 017400 104406

40\$:  
1\$:  
3\$:

CLR R2  
BIS #FC,CCR  
BIT #VCIP,CCR  
BEQ 10\$  
DEC R2  
BNE 3\$

;INITIALIZE COUNTER  
;START FLUSH  
;SEE IF FLUSH COMPLETE  
;FLUSH COMPLETE  
;SEE IF TIME HAS RUN OUT  
;NOT YET  
;INSTRUCTION 'JMP 1\$' PLACED HERE  
;FOR LOOP ON ERROR  
;ERROR  
;-----

017402 017400

.WORD -2

2990  
2991  
2992

;FLUSH CACHE TESTS  
;FLUS IN PROGRESS BIT FAILED TO CLEAR  
;TIME FOR FLUSH TO COMPLETE RAN OUT

2993 017404 000000  
2994 017406 000240  
017410 005237 001472

10\$:

.WORD 0  
NOP  
INC \$TESTN

;END OF TEST  
;INCREMENT TEST COUNTER

2999

.SBTTL TEST # 121 - CHECK THAT VSIU BIT SETS  
:\*\*\*\*\*  
:\*TEST 121 - CHECK THAT VSIU BIT SETS  
:\* VERIFY THAT VSIU BIT WILL CHANGE FROM A CLEAR TO SET CONDITION AS  
:\* A RESULT OF CACHE FLUSH  
:\*\*\*\*\*

3000 017414 000004

TST121: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION

017416 017426  
017420 017426  
017422 000000  
017424 017472  
017426

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

3001 017426 032737 020000 177746

40\$: 1\$: BIT #VSIU,CCR ;IS SET A BEING USED

3002 017434 001407

BEQ 3\$ ;YES

3003 017436 052737 000400 177746

BIS #FC,CCR ;CAUSE FLUSH FOR SET A

3004 017444 032737 010000 177746

200\$: BIT #VCIP,CCR ;WAIT FOR FLUSH TO COMPLETE

3005 017452 001374

BNE 200\$

3006 017454 052737 000400 177746

3\$: BIS #FC,CCR ;CAUSE FLUSH

3007 017462 032737 010000 177746

4\$: BIT #VCIP,CCR ;WAIT FOR FLUSH TO COMPLETE

3008 017470 001374

BNE 4\$

3009 017472 000240

25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE

017474 000240

NOP ;FOR LOOP ON ERROR

3010 017476 032737 020000 177746

BIT #VSIU,CCR ;IS VSIU BIT =1 INDICATING VALID SET

3011

;B WAS SELECTED

3012 017504 001003

BNE 10\$ ;PASS

3013 017506 104406

ERROR ;ERROR

017510 017506

.WORD .-2 ;-----

3014

;FLUSH CACHE TESTS

3015

;VSIU BIT DID NOT SET AS A RESULT OF FLUSH

3016 017512 000000

.WORD 0

3017 017514 000240

10\$: NOP ;END OF TEST

017516 005237 001472

INC \$TESTN ;INCREMENT TEST COUNTER

3022

.SBTTL TEST # 122 - CHECK THAT VSIU BIT CLEARS  
:\*\*\*\*\*  
:TEST 122 - CHECK THAT VSIU BIT CLEARS  
:\* VERIFY THAT VSIU BIT WILL CHANGE FROM A SET TO CLEAR CONDITION AS  
:\* A RESULT OF CACHE FLUSH  
:\*\*\*\*\* \*

3023 017522 000004

TST122:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION

017524 017534  
017526 017534  
017530 000000  
017532 017600  
017534

.WORD 40\$  
.WORD 1\$  
.WORD 0  
.WORD 25\$

3024 017534 032737 020000 177746

40\$: BIT #VSIU,CCR ;IS SET B BEING USED  
1\$: BNE 3\$ ;YES

3025 017542 001007

BIS #FC,CCR ;CAUSE FLUSH FOR SET B

3026 017544 052737 000400 177746

200\$: BIT #VCIP,CCR ;WAIT FOR FLUSH TO COMPLETE  
BNE 200\$

3027 017552 032737 010000 177746

3\$: BIS #FC,CCR ;CAUSE FLUSH  
4\$: BIT #VCIP,CCR ;WAIT FOR FLUSH TO COMPLETE  
BNE 4\$

3028 017560 001374

25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR

3029 017562 052737 000400 177746

BIT #VSIU,CCR ;IS VSIU BIT =0 INDICATING VALID SET  
A WAS SELECTED

3030 017570 032737 010000 177746

BEQ 10\$ ;PASS  
ERROR ;ERROR  
;-----

3031 017576 001374

.WORD -2

3032 017600 000240

;FLUSH CACHE TESTS  
;VSIU BIT DID NOT CLEAR AS A RESULT OF FLUSH

3033 017604 032737 020000 177746

.WORD 0

3034 017612 001403

10\$: NOP ;END OF TEST  
INC \$TESTN ;INCREMENT TEST COUNTER

3035 017614 104406

017616 017614

3037

3038

3039 017620 000000

3040 017622 000240

017624 005237 001472

CKKKABO 11-44 KK11B CACHE  
TEST # 123 - WRITE AND READ 0'S TO ALL LOW CACHE VALID  
3046

MACRO M1113 27-OCT-80 08:39 PAGE 104 SEQUENCE 123

.SBTTL TEST # 123 - WRITE AND READ 0'S TO ALL LOW CACHE VALID  
\*\*\*\*\*  
:TEST 123 - WRITE AND READ 0'S TO ALL LOW CACHE VALID  
:WRITE AND READ 0'S TO ALL LOW CACHE VALID  
:BIT STORE ADDRESS LOCATIONS- SET A  
:(VALID STORE LOCATIONS 0000 TO 3777)  
\*\*\*\*\*

017630  
3047 017630 000004  
  
017632 017642  
017634 070052  
017636 000000  
017640 070100  
017642 012737 001015 177746 40\$:  
017650 004437 002370  
017654 020042

TST123:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
.WORD 40\$ ;LOOP ON ERROR START LOCATION  
.WORD 1\$-40\$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 0 ;LOOP ON ERROR END LOCATION  
.WORD 25\$-40\$+67764 ;DISABLE CACHE  
MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE  
JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST  
.WORD 10\$+2

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

3048 017656 032737 020000 177746  
3049 017664 001407  
3050 017666 052737 000400 177746  
3051 017674 032737 010000 177746 200\$:  
3052 017702 001374  
3053 017704 012737 177777 177752 3\$:  
3054  
3055 017712 112737 000374 177751  
3056 017720 012705 060000  
3057 017724 012703 040000  
3058 017730 012737 000015 177746 1\$:  
3059 017736 112737 000003 177750  
3060  
3061  
3062  
3063  
3064  
3065 017744 005713  
3066 017746 005715  
3067  
3068 017750 005715  
3069  
3070  
3071 017752 013701 177750  
3072 017756 000240 25\$:  
017760 000240  
3073 017762 105037 177750  
3074 017766 012737 001015 177746  
3075 017774 032701 010000  
3076 020000 001410  
3077 020002 010537 047334  
3078  
3079 020006 006237 047334  
3080 020012 104406  
  
020014 020012

BIT #VSIU,CCR ;IS SET A BEING USED?  
BEQ 3\$ ;YES  
BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A  
3051 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
BNE 200\$  
3053 MOV #-1,CHR ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR  
;REGISTERS, SINCE TDAR WILL BE USED  
3055 MOV #374,CMR+1  
3056 MOV #60000,R5 ;:ADDRESS 60000 INTO R5  
3057 MOV #40000,R3 ;:ADDRESS 40000 INTO R3  
3058 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE  
3059 MOV #HODO+TDAR,CMR ;HODO ALLOWS VALID STORE SET A TO  
;BE WRITTEN TO CMR<12> ONLY DURING  
;THE DESTINATION MEMORY ACCESS.  
;TDAR WILL FORCE A 0 TO BE WRITTEN  
;INTO VALID STORE WHEN A WRITE TO  
;VALID STORE OCCURS  
3065 TST (R3)  
3066 TST (R5) ;WRITE A 0 INTO VALID STORE ADDRESS  
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.  
3068 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>  
;FROM VALID STORE ADDRESS LOCATION  
;JUST WRITTEN INTO.  
3071 MOV CMR,R1 ;SAVE CMR DATA  
3072 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
3073 CLRB CMR ;DISABLE MAINT. MODE  
3074 MOV #OFF,CCR ;DISABLE CACHE  
3075 BIT #VLD,R1 ;CMR<12> SHOULD BE 0.  
3076 BEQ 9\$ ;PASS  
3077 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION  
;USED: CA<12:1>  
3079 ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
3080 ERROR ;ERROR  
;-----  
;-----  
.WORD -2

```

3081
3082
3083
3084 020016 047334
3085
3086 020020 000000
3087 020022 062705 000002
3088 020026 062703 000002
3089 020032 020527 070000
3090 020036 001334
3091 020040 000240
      020042 005237 001472

```

```

          CA121
          .WORD 0
9$:      ADD #2,R5
          ADD #2,R3
          CMP R5,#70000
          BNE 1$
10$:     NOP
          INC $TESTN

```

```

;VALID BITS STORE TESTS
;READING VALID STORE DATA SET A
;THRU CMR<12> DID NOT RESULT IN 0.
;PRINT VALID STORE ADDRESS LOCATION
;USED: CA<12:1>.

;NEXT VALID STORE LOCATION

;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
:NO
;END OF TEST
;INCREMENT TEST COUNTER

```

TEST # 124 - WRITE AND READ 1'S TO ALL LOW CACHE VALID

3097

.SBTTL TEST # 124 - WRITE AND READ 1'S TO ALL LOW CACHE VALID

\*\*\*\*\*
\*TEST 124 - WRITE AND READ 1'S TO ALL LOW CACHE VALID
\* WRITE AND READ 1'S TO ALL LOW CACHE VALID
\* BIT STORE ADDRESS LOCATIONS- SET A
\* (VALID STORE LOCATIONS 0000 TO 3777)
\*\*\*\*\*

TST124:

3098 020046 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
020050 020060 .WORD 40\$ ;TEST START LOCATION
020052 070036 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION
020054 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
020056 070064 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION
020060 012737 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE
020066 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
020072 020244 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$
;ARE RELOCATED TO HI CACHE SPACE

3099 020074 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
3100 020102 001407 BEQ 3\$ ;YES
3101 020104 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A
3102 020112 032737 010000 177746 200\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3103 020120 001374 BNE 200\$
3104 020122 012705 060000 3\$: MOV #60000,R5 ;:ADDRESS 60000 INTO R5
3105 020126 012703 040000 MOV #40000,R3 ;:ADDRESS 40000 INTO R3
3106 020132 012737 000015 177746 1\$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3107 020140 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS VALID STORE SET A TO
3108 ;BE WRITTEN TO CMR<12> ONLY DURING
3109 ;THE DESTINATION MEMORY ACCESS.
3110 020146 005713 TST (R3)
3111 020150 005715 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS
3112 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3113 020152 005715 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
3114 ;FROM VALID STORE ADDRESS LOCATION
3115 ;JUST WRITTEN INTO.
3116 020154 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
3117 020160 000240 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
020162 000240 NOP ;FOR LOOP ON ERROR
3118 020164 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
3119 020170 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3120 020176 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.
3121 020202 001010 BNE 9\$ ;PASS
3122
3123 020204 010537 047334 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
3124 ;USED: CA<12:1>
3125 020210 006237 047334 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3126 020214 104406 ERROR ;ERROR
;-----
020216 020214 .WORD -2
3127 ;VALID BITS STORE TESTS
3128 ;READING VALID STORE DATA SET A
3129 ;THRU CMR<12> DID NOT RESULT IN 1.
3130 020220 047334 CA121 ;PRINT VALID STORE ADDRESS LOCATION
3131 ;USED: CA<12:1>.

3132	020222	000000			.WORD	0	
3133	020224	062705	000002	9\$:	ADD	#2,R5	;NEXT VALID STORE LOCATION
3134	020230	062703	000002		ADD	#2,R3	
3135	020234	020527	070000		CMP	R5,#70000	;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
3136	020240	001334			BNE	1\$	;NO
3137	020242	000240		10\$:	NOP		;END OF TEST
	020244	005237	001472		INC	\$TESTN	;INCREMENT TEST COUNTER



CKKKABO 11-44 KK11B CACHE  
TEST # 125 - WRITE AND READ 0'S  
3143

MACRO M1113 27-OCT-80 08:39 PAGE 106 SEQUENCE 127  
TO ALL HIGH CACHE VALID

K 10

.SBTTL TEST # 125 - WRITE AND READ 0'S TO ALL HIGH CACHE VALID  
\*\*\*\*\*  
\*TEST 125 - WRITE AND READ 0'S TO ALL HIGH CACHE VALID  
\* WRITE AND READ 0'S TO ALL HIGH CACHE VALID  
\* BIT STORE ADDRESS LOCATIONS- SET A  
\* (VALID STORE LOCATIONS 4000 TO 7777)  
\*\*\*\*\*

3144 020250 000004  
020252 020262  
020254 060052  
020256 000000  
020260 060100  
020262 012737 001015 177746  
020270 004437 002342  
020274 020462

TST125:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO LOW CACHE SPACE

3145 020276 032737 020000 177746  
3146 020304 001407  
3147 020306 052737 000400 177746  
3148 020314 032737 010000 177746  
3149 020322 001374  
3150 020324 012737 177777 177752  
3151  
3152 020332 112737 000374 177751  
3153 020340 012705 070000  
3154 020344 012703 050000  
3155 020350 012737 000015 177746  
3156 020356 112737 000003 177750  
3157  
3158  
3159  
3160  
3161  
3162 020364 005713  
3163 020366 005715  
3164  
3165 020370 005715  
3166  
3167  
3168 020372 013701 177750  
3169 020376 000240  
020400 000240  
3170 020402 105037 177750  
3171 020406 012737 001015 177746  
3172 020414 032701 010000  
3173 020420 001410  
3174 020422 010537 047334  
3175  
3176 020426 006237 047334  
3177 020432 104406  
020434 020432

BIT #VSIU,CCR ;IS SET A BEING USED?  
BEQ 3\$ ;YES  
BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A  
200\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
BNE 200\$  
3\$: MOV #-1,CHR ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR  
;REGISTERS, SINCE TDAR WILL BE USED  
MOV #374,CMR+1  
MOV #70000,R5 ;:ADDRESS 70000 INTO R5  
MOV #50000,R3 ;:ADDRESS 50000 INTO R3  
1\$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE  
MOV #HODO+TDAR,CMR ;HODO ALLOWS VALID STORE SET A TO  
;BE WRITTEN TO CMR<12> ONLY DURING  
;THE DESTINATION MEMORY ACCESS.  
;TDAR WILL FORCE A 0 TO BE WRITTEN  
;INTO VALID STORE WHEN A WRITE TO  
;VALID STORE OCCURS  
TST (R3)  
TST (R5) ;WRITE A 0 INTO VALID STORE ADDRESS  
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.  
TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>  
;FROM VALID STORE ADDRESS LOCATION  
;JUST WRITTEN INTO.  
MOV CMR,R1 ;SAVE CMR DATA  
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINT. MODE  
MOV #OFF,CCR ;DISABLE CACHE  
BIT #VLD,R1 ;CMR<12> SHOULD BE 0.  
BEQ 9\$ ;PASS  
MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION  
;USED: CA<12:1>  
ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
ERROR ;ERROR  
;-----  
.WORD -2

3178  
3179  
3180  
3181 020436 047334  
3182  
3183 020440 000000  
3184 020442 062705 000002  
3185 020446 062703 000002  
3186 020452 020527 100000  
3187 020456 001334  
3188 020460 000240  
          020462 005237 001472

CA121  
          .WORD 0  
9\$: ADD #2,R5  
      ADD #2,R3  
      CMP R5,#100000  
      BNE 1\$  
10\$: NOP  
      INC \$TESTN

:VALID BITS STORE TESTS  
:READING VALID STORE DATA SET A  
:THRU CMR<12> DID NOT RESULT IN 0.  
:PRINT VALID STORE ADDRESS LOCATION  
:USED: CA<12:1>.  
  
:NEXT VALID STORE LOCATION  
  
:HAVE ALL HIGH CACHE ADDRESS LOCATIONS BEEN DONE?  
:NO  
:END OF TEST  
:INCREMENT TEST COUNTER

3194

.SBTTL TEST # 126 - WRITE AND READ 1'S TO ALL HIGH CACHE VALID  
 :\*\*\*\*\*  
 :\*TEST 126 - WRITE AND READ 1'S TO ALL HIGH CACHE VALID  
 :\* WRITE AND READ 1'S TO ALL HIGH CACHE VALID  
 :\* BIT STORE ADDRESS LOCATIONS- SET A  
 :\* (VALID STORE LOCATIONS 4000 TO 7777)  
 :\*\*\*\*\*

020466  
 3195 020466 000004  
 020470 020500  
 020472 060036  
 020474 000000  
 020476 060064  
 020500 012737 001015 177746  
 020506 004437 002342  
 020512 020664

TST126:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
 40\$: MOV #OFF,CCR ;DISABLE CACHE  
 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
 :ARE RELOCATED TO LOW CACHE SPACE

3196 020514 032737 020000 177746  
 3197 020522 001407  
 3198 020524 052737 000400 177746  
 3199 020532 032737 010000 177746  
 3200 020540 001374  
 3201 020542 012705 070000  
 3202 020546 012703 050000  
 3203 020552 012737 000015 177746  
 3204 020560 112737 000002 177750  
 3205  
 3206  
 3207 020566 005713  
 3208 020570 005715  
 3209  
 3210 020572 005715  
 3211  
 3212  
 3213 020574 013701 177750  
 3214 020600 000240  
 020602 000240  
 3215 020604 105037 177750  
 3216 020610 012737 001015 177746  
 3217 020616 032701 010000  
 3218 020622 001010  
 3219 020624 010537 047334  
 3220  
 3221 020630 006237 047334  
 3222 020634 104406  
 020636 020634  
 3223  
 3224  
 3225  
 3226 020640 047334  
 3227  
 3228 020642 000000

BIT #VSIU,CCR ;IS SET A BEING USED?  
 BEQ 3\$ ;YES  
 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A  
 200\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
 BNE 200\$  
 3\$: MOV #70000,R5 ;:ADDRESS 70000 INTO R5  
 MOV #50000,R3 ;:ADDRESS 50000 INTO R3  
 1\$: MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE  
 MOVB #HODO,CMR ;:HODO ALLOWS VALID STORE SET A TO  
 ;:BE WRITTEN TO CMR<12> ONLY DURING  
 ;:THE DESTINATION MEMORY ACCESS.  
 TST (R3)  
 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS  
 ;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.  
 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>  
 ;:FROM VALID STORE ADDRESS LOCATION  
 ;:JUST WRITTEN INTO.  
 25\$: MOV CMR,R1 ;SAVE CMR DATA  
 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP ;FOR LOOP ON ERROR  
 CLRB CMR ;DISABLE MAINT. MODE  
 MOV #OFF,CCR ;DISABLE CACHE  
 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.  
 BNE 9\$ ;PASS  
 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION  
 ;:USED: CA<12:1>  
 ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
 ERROR ;ERROR  
 ;-----  
 .WORD -2 ;VALID BITS STORE TESTS  
 ;:READING VALID STORE DATA SET A  
 ;:THRU CMR<12> DID NOT RESULT IN 0.  
 CA121 ;PRINT VALID STORE ADDRESS LOCATION  
 ;:USED: CA<12:1>.  
 .WORD 0

3229	020644	062705	000002	9\$:	ADD	#2,R5	;NEXT VALID STORE LOCATION
3230	020650	062703	000002		ADD	#2,R3	
3231	020654	020527	100000		CMP	R5,#100000	;HAVE ALL HIGH CACHE ADDRESS LOCATIONS BEEN DONE?
3232	020660	001334			BNE	1\$	;NO
3233	020662	000240		10\$:	NOP		;END OF TEST
	020664	005237	001472		INC	\$TESTN	;INCREMENT TEST COUNTER

TEST # 127 - VERIFY VALID DATA STORE ADRS LINES (CA(12:1))  
3237

.SBTTL TEST # 127 - VERIFY VALID DATA STORE ADRS LINES (CA(12:1))  
:\*\*\*\*\*  
:TEST 127 - VERIFY VALID DATA STORE ADRS LINES (CA(12:1))  
:\* VERIFY VALID DATA STORE ADDRESS LINES (CA<12:1>)  
:\*\*\*\*\*

TST127:

3238	020670	000004				SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON :ERROR/LOOP ON TEST :TEST START LOCATION
	020672	020702				.WORD	40\$	:LOOP ON ERROR START LOCATION
	020674	070054				.WORD	1\$-40\$+67764	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	020676	000000				.WORD	0	:LOOP ON ERROR END LOCATION
	020700	070130				.WORD	25\$-40\$+67764	:DISABLE CACHE
	020702	012737	001015	177746	40\$:	MOV	#OFF,CCR	:LOCATE TEST CODE TO HIGH CACHE SPACE
	020710	004437	002370			JSR	R4,RELCTH	:ADDRESS OF START OF NEXT TEST
	020714	021132				.WORD	10\$+2	

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

3239	020716	032737	020000	177746		BIT	#VSIU,CCR	:IS SET A BEING USED?
3240	020724	001407				BEQ	3\$	:YES
3241	020726	052737	000400	177746		BIS	#FC,CCR	:NO; FLUSH CACHE FOR SET A
3242	020734	032737	010000	177746	4\$:	BIT	#VCIP,CCR	:WAIT TILL FLUSH COMPLETE
3243	020742	001374				BNE	4\$	
3244	020744	012737	000002	047344	3\$:	MOV	#2,FLTPAT	:1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
3245	020752	012702	040000		2\$:	MOV	#40000,R2	
3246	020756	012703	060000			MOV	#60000,R3	
3247	020762	063702	047344			ADD	FLTPAT,R2	:R2 CONTAINS 40000+FLTPAT
3248	020766	063703	047344			ADD	FLTPAT,R3	:R3 CONTAINS 60000+FLTPAT
3249	020772	112737	000002	177750	1\$:	MOVB	#HODO,CMR	:HODO ALLOWS VALID DATA STORE BITS TO BE :WRITTEN TO CMR<12> ONLY DURING THE :DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3250								:NO UCB SO AS TO WRITE ENABLE VALID STORE
3251								:TDAR WILL FORCE A 0 TO BE WRITTEN :INTO VALID STORE WHEN A WRITE TO :VALID STORE OCCURS.
3252	021000	012737	000015	177746		MOV	#15,CCR	
3253	021006	152737	000001	177750		BISB	#TDAR,CMR	
3254								
3255								
3256	021014	005737	040000			TST	40000	
3257	021020	005737	060000			TST	60000	:WRITE 0 INTO VALID STORE LOCATION 0000.
3258	021024	142737	000001	177750		BICB	#TDAR,CMR	:CLEARING TDAR WILL ALLOW A 1 TO BE :WRITTEN INTO VALID STORE WHEN A WRITE :TO VALID STORE OCCURS.
3259								
3260								
3261	021032	005712				TST	(R2)	
3262	021034	005713				TST	(R3)	:WRITE 1 INTO VALID STORE LOCATION :SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
3263								:LOAD DATA FROM VALID DATA STORE LOCATION :0000 INTO CMR<12>.
3264	021036	005737	060000			TST	60000	
3265								
3266	021042	013701	177750			MOV	CMR,R1	:SAVE CMR DATA
3267	021046	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	021050	000240				NOP		:FOR LOOP ON ERROR
3268	021052	105037	177750			CLRB	CMR	:DISABLE MAINT. MODE
3269	021056	012737	001015	177746		MOV	#OFF,CCR	:DISABLE CACHE
3270	021064	032701	010000			BIT	#VLD,R1	:CMR<12> SHOULD READ 0.
3271	021070	001411				BEQ	9\$	:PASS
3272								:ROUTINE
3273	021072	013737	047344	047334		MOV	FLTPAT,CA121	:SAVE CA<12:1> USED
3274	021100	006237	047334			ASR	CA121	:PREPARE CA121 FOR TYPEOUT
3275	021104	104406				ERROR		:ERROR

021106 021104

.WORD .-2

;-----

;VALID STOR ADDRESS VERIFICATION.  
 ;VALID STORE LOCATION 0000 DID NOT  
 ;READ AS A 0 INDICATING THAT IT WAS  
 ;OVERWRITTEN WITH A 1. THIS SUGGESTS  
 ;A BAD CA<12:1> VALID STORE ADDRESS LINE.  
 ;PRINT VALID STORE ADDRESS FAILURE: CA<12:1>.  
 ;NOTE THAT THE 1 IN THIS PATTERN  
 ;WILL POINT TO THE ADDRESS LINE OF  
 ;THAT BROUGHT OUT ERROR.

3276  
 3277  
 3278  
 3279  
 3280  
 3281  
 3282  
 3283  
 3284  
 3285  
 3286  
 3287  
 3288  
 3289

021110 047334

CA121

021112 000000

.WORD 0

021114 006337 047344

9\$:

ASL FLTPAT

;NEXT PATTERN

021120 022737 020000 047344

CMP #20000,FLTPAT

;HAS VALID DATA STORE ADDRESS 4000 BEEN DONE?

021126 001311

BNE 2\$

;NO

021130 000240

10\$:

NOP

;END OF TEST

021132 005237 001472

INC \$TESTN

;INCREMENT TEST COUNTER

TEST # 130 - LOW CACHE INVALIDATE WITH CACHE FLUSH

3295

.SBTTL TEST # 130 - LOW CACHE INVALIDATE WITH CACHE FLUSH
\*\*\*\*\*
\*TEST 130 - LOW CACHE INVALIDATE WITH CACHE FLUSH
\* VERIFY THAT ALL LOW CACHE VALID STORE SET A ADDRESS LOCATIONS
\* WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
\* (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 0000-3777)
\*\*\*\*\*

021136
3296 021136 000004
021140 021150
021142 070074
021144 000000
021146 070130
021150 012737 001015 177746 40\$:
021156 004437 002370
021162 021410

TST130:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
.WORD 40\$ ;TEST START LOCATION
.WORD 18-40\$+67764 ;LOOP ON ERROR START LOCATION
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$
:ARE RELOCATED TO HI CACHE SPACE

3297 021164 012705 060000
3298 021170 012703 040000
3299 021174 032737 020000 177746
3300 021202 001407
3301 021204 052737 000400 177746
3302 021212 032737 010000 177746 200\$:
3303 021220 001374
3304 021222 012737 000015 177746 3\$:
3305 021230 112737 000002 177750
3306
3307
3308 021236 005713 4\$:
3309 021240 005715
3310
3311 021242 062705 000002
3312 021246 062703 000002
3313 021252 020527 070000
3314 021256 001367
3315 021260 052737 000400 177746 1\$:
3316
3317 021266 032737 010000 177746 500\$:
3318 021274 001374
3319 021276 052737 000400 177746
3320 021304 032737 010000 177746 6\$:
3321 021312 001374
3322 021314 000240 25\$:
021316 000240
3323 021320 012705 060000
3324 021324 005715 2\$:
3325
3326
3327 021326 013701 177750
3328 021332 032701 010000
3329 021336 001416
3330 021340 105037 177750
3331 021344 012737 001015 177746

MOV #60000,R5 ;:ADDRESS 60000 INTO R5
MOV #40000,R3 ;:ADDRESS 40000 INTO R3
BIT #VSIU,CCR ;:IS SET A BEING USED?
BEQ 3\$ ;:YES
BIS #FC,CCR ;:NO; FLUSH CACHE FOR SET A
BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
BNE 200\$
MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
MOVB #HODO,CMR ;:HODO ALLOWS VALID STORE SET A TO
;:BE WRITTEN TO CMR<12> ONLY DURING
;:THE DESTINATION MEMORY ACCESS.
TST (R3) 4\$:
TST (R5) ;:WRITE A 1 INTO VALID STORE ADDRESS
;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
;:NEXT VALID STORE LOCATION
ADD #2,R5
ADD #2,R3
CMP R5,#70000 ;:HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
BNE 4\$ ;:NO
BIS #FC,CCR ;:FLUSH CACHE TO SELECT SET B AND
;:INVALIDATE SET A
BIT #VCIP,CCR 500\$: ;:WAIT TILL FLUSH COMPLETE
BNE 500\$
BIS #FC,CCR ;:FLUSH TO SELECT SET A AGAIN
BIT #VCIP,CCR 6\$: ;:WAIT
BNE 6\$
NOP 25\$: ;:INSTRUCTION 'JMP 1\$' PLACED HERE
NOP ;:FOR LOOP ON ERROR
MOV #60000,R5 ;:ADDRESS 60000 INTO R5
TST (R5) 2\$: ;:WRITE VALID STORE DATA INTO CMR<12>
;:FROM ADDRESS LOCATION SPECIFIED BY
;:R5'S BITS 12-1.
MOV CMR,R1 ;:SAVE CMR DATA
BIT #VLD,R1 ;:CMR<12> SHOULD BE 0
BEQ 9\$ ;:PASS
CLRB CMR ;:DISABLE MAINT. MODE
MOV #OFF,CCR ;:DISABLE CACHE

3332	021352	010537	047342		MOV	R5,CNT121		;SAVE VALID STORE FLUSH ADDRESS LOCATION
3333								;USED: CNT<12:1>
3334	021356	006237	047342		ASR	CNT121		;PREPARE CNT121 FOR TYPEOUT
3335	021362	104406			ERROR			;ERROR
								;-----
	021364	021362			.WORD	.-2		
3336								;FLUSH CACHE INVALID TEST-SET A
3337								;READING VALID STORE LOCATION FROM SET A THRU CMR<12>
3338								;DID NOT RESULT IN A ZERO,INDICATING THAT
3339								;THE CACHE FLUSH DID NOT INVALIDATE THIS
3340								; LOCATION.
3341	021366	047342			CNT121			;PRINT VALID STORE FLUSH ADDRESS LOCATION
3342								;IN ERROR: CNT<12:1>.
3343	021370	000000			.WORD	0		
3344	021372	000405			BR	10\$		;IF ERROR, END TEST
3345	021374	062705	000002	9\$:	ADD	#2,R5		;NEXT VALID STORE LOCATION
3346	021400	020527	070000		CMP	R5,#70000		;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
3347	021404	001347			BNE	2\$		;NO
3348	021406	000240		10\$:	NOP			;END OF TEST
	021410	005237	001472		INC	\$TESTN		;INCREMENT TEST COUNTER



TEST # 131 - HIGH CACHE INVALIDATE WITH CACHE FLUSH

3354

.SBTTL TEST # 131 - HIGH CACHE INVALIDATE WITH CACHE FLUSH

\*\*\*\*\*

\*TEST 131 - HIGH CACHE INVALIDATE WITH CACHE FLUSH

\* VERIFY THAT ALL HI CACHE VALID STORE SET A ADDRESS LOCATIONS

\* WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.

\* (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 4000-7777)

\*\*\*\*\*

TST131:

3355	021414	000004		SCPCND		;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
						;ERROR/LOOP ON TEST
	G21416	021426		.WORD	40\$	;TEST START LOCATION
	021420	060074		.WORD	1\$-40\$+57764	;LOOP ON ERROR START LOCATION
	021422	000000		.WORD	0	;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	021424	060130		.WORD	25\$-40\$+57764	;LOOP ON ERROR END LOCATION
	021426	012737	001015	MOV	#OFF,CCR	;DISABLE CACHE
	021434	004437	002342	JSR	R4,RELCTL	;LOCATE TEST CODE TO LOW CACHE SPACE
	021440	021666		.WORD	10\$+2	;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$

;ARE RELOCATED TO LOW CACHE SPACE

3356	021442	012705	070000		MOV	#70000,R5	::ADDRESS 70000 INTO R5
3357	021446	012703	050000		MOV	#50000,R3	;ADDRESS 50000 INTO R3
3358	021452	032737	020000	177746	BIT	#VSIU,CCR	;IS SET A BEING USED?
3359	021460	001407			BEQ	3\$	;YES
3360	021462	052737	000400	177746	BIS	#FC,CCR	;NO; FLUSH CACHE FOR SET A
3361	021470	032737	010000	177746	200\$:	BIT	#VCIP,CCR
3362	021476	001374			BNE	200\$	;WAIT TILL FLUSH COMPLETE
3363	021500	012737	000015	177746	3\$:	MOV	#15,CCR
3364	021506	112737	000002	177750	MOV	#HODO,CMR	;NO UCB SO AS TO WRITE ENABLE VALID STORE
3365							;HODO ALLOWS VALID STORE SET A TO
3366							;BE WRITTEN TO CMR<12> ONLY DURING
3367	021514	005713			4\$:	TST	(R3)
3368	021516	005715				TST	(R5)
3369							;WRITE A 1 INTO VALID STORE ADDRESS
3370	021520	062705	000002			ADD	#2,R5
3371	021524	062703	000002			ADD	#2,R3
3372	021530	020527	100000			CMP	R5,#100000
3373	021534	001367				BNE	4\$
3374	021536	052737	000400	177746	1\$:	BIS	#FC,CCR
3375							;HAVE ALL HI CACHE LOCATIONS BEEN DONE?
3376	021544	032737	010000	177746	500\$:	BIT	#VCIP,CCR
3377	021552	001374				BNE	500\$
3378	021554	052737	000400	177746		BIS	#FC,CCR
3379	021562	032737	010000	177746	6\$:	BIT	#VCIP,CCR
3380	021570	001374				BNE	6\$
3381	021572	000240			25\$:	NOP	
	021574	000240				NOP	
3382	021576	012705	070000			MOV	#70000,R5
3383	021602	005715			2\$:	TST	(R5)
3384							;INSTRUCTION 'JMP 1\$' PLACED HERE
3385							;FOR LOOP ON ERROR
3386	021604	013701	177750			MOV	CMR,R1
3387	021610	032701	010000			BIT	#VLD,R1
3388	021614	001416				BEQ	9\$
3389	021616	105037	177750			CLRB	CMR
3390	021622	012737	001015	177746		MOV	#OFF,CCR

;DISABLE MAINT. MODE

;DISABLE CACHE



3413

.SBTTL TEST # 132 - WRITE AND READ 0'S TO ALL LOW CACHE VALID  
 \*\*\*\*\*  
 \*TEST 132 - WRITE AND READ 0'S TO ALL LOW CACHE VALID  
 \* WRITE AND READ 0'S TO ALL LOW CACHE VALID  
 \* BIT STORE ADDRESS LOCATIONS- SET B  
 \* (VALID STORE LOCATIONS 0000 TO 3777)  
 \*\*\*\*\*

3414	021672	000004				TST132:	SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON :ERROR/LOOP ON TEST
	021674	021704					.WORD 40\$		:TEST START LOCATION
	021676	070052					.WORD 1\$-40\$+67764		:LOOP ON ERROR START LOCATION
	021700	000000					.WORD 0		:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	021702	0701C0					.WORD 25\$-40\$+67764		:LOOP ON ERROR END LOCATION
	021704	012737	001015	177746	40\$:		MOV #OFF,CCR		:DISABLE CACHE
	021712	004437	002370				JSR R4,RELCTH		:LOCATE TEST CODE TO HIGH CACHE SPACE
	021716	022104					.WORD 10\$+2		:ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
 :ARE RELOCATED TO HI CACHE SPACE

3415	021720	032737	020000	177746			BIT #VSIU,CCR		:IS SET B BEING USED?
3416	021726	001007					BNE 3\$		:YES
3417	021730	052737	000400	177746			BIS #FC,CCR		:NO; FLUSH CACHE FOR SET B
3418	021736	032737	010000	177746	200\$:		BIT #VCIP,CCR		:WAIT TILL FLUSH COMPLETE
3419	021744	001374					BNE 200\$		
3420	021746	012737	177777	177752	3\$:		MOV #-1,CHR		:LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR :REGISTERS, SINCE TDAR WILL BE USED
3421									
3422	021754	112737	000374	177751			MOVB #374,CMR+1		
3423	021762	012705	060000				.V #60000,R5		::ADDRESS 60000 INTO R5
3424	021766	012703	040000				MOV #40000,R3		:ADDRESS 40000 INTO R3
3425	021772	012737	000015	177746	1\$:		MOV #15,CCR		:NO UCB SO AS TO WRITE ENABLE VALID STORE
3426	022000	112737	000003	177750			MOVB #HODO+TDAR,CMR		:HODO ALLOWS VALID STORE SET B TO :BE WRITTEN TO CMR<12> ONLY DURING :THE DESTINATION MEMORY ACCESS. :TDAR WILL FORCE A 0 TO BE WRITTEN :INTO VALID STORE WHEN A WRITE TO :VALID STORE OCCURS
3427									
3428									
3429									
3430									
3431									
3432	022006	005713					TST (R3)		
3433	022010	005715					TST (R5)		:WRITE A 0 INTO VALID STORE ADDRESS :LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3434									
3435	022012	005715					TST (R5)		:WRITE VALID STORE DATA INTO CMR<12> :FROM VALID STORE ADDRESS LOCATION :JUST WRITTEN INTO.
3436									
3437									
3438	022014	013701	177750				MOV CMR,R1		:SAVE CMR DATA
3439	022020	000240			25\$:		NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	022022	000240					NOP		:FOR LOOP ON ERROR
3440	022024	105037	177750				CLRB CMR		:DISABLE MAINT. MODE
3441	022030	012737	001015	177746			MOV #OFF,CCR		:DISABLE CACHE
3442	022036	032701	010000				BIT #VLD,R1		:CMR<12> SHOULD BE 0.
3443	022042	001410					BEQ 9\$		:PASS
3444	022044	010537	047334				MOV R5,CA121		:SAVE VALID STORE ADDRESS LOCATION
3445									:USED: CA<12:1>
3446	022050	006237	047334				ASR CA121		:PREPARE CA121 FOR TYPEOUT
3447	022054	104406					ERROR		:ERROR
									:-----
	022056	022054					.WORD -2		

3448  
3449  
3450  
3451 022060 047334  
3452  
3453 022062 000000  
3454 022064 062705 000002  
3455 022070 062703 000002  
3456 022074 020527 070000  
3457 022100 001334  
3458 022102 000240  
022104 005237 001472

CA121  
9\$: .WORD 0  
ADD #2,R5  
ADD #2,R3  
CMP R5,#70000  
BNE 1\$  
10\$: NOP  
INC \$TESTN

;VALID BITS STORE TESTS-SET B  
;READING VALID STORE DATA SET B  
;THRU CMR<12> DID NOT RESULT IN 0.  
;PRINT VALID STORE ADDRESS LOCATION  
;USED: CA<12:1>.  
;NEXT VALID STORE LOCATION  
;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?  
;NO  
;END OF TEST  
;INCREMENT TEST COUNTER

J 11  
 .SBTTL TEST # 133 - WRITE AND READ 1'S TO ALL LOW CACHE VALID  
 \*\*\*\*\*  
 \*TEST 133 - WRITE AND READ 1'S TO ALL LOW CACHE VALID  
 \* WRITE AND READ 1'S TO ALL LOW CACHE VALID  
 \* BIT STORE ADDRESS LOCATIONS- SET B  
 \* (VALID STORE LOCATIONS 0000 TO 3777)  
 \*\*\*\*\*

022110  
 3465 022110 000004  
 022112 022122  
 022114 070036  
 022116 000000  
 022120 070064  
 022122 012737 001015 177746 40\$:  
 022130 004437 002370  
 022134 022306

TST133:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
 MOV #OFF,CCR ;DISABLE CACHE  
 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

3466 022136 032737 020000 177746  
 3467 022144 001007  
 3468 022146 052737 000400 177746  
 3469 022154 032737 010000 177746 200\$:  
 3470 022162 001374  
 3471 022164 012705 060000 3\$:  
 3472 022170 012703 040000  
 3473 022174 012737 000015 177746 1\$:  
 3474 022202 112737 000002 177750  
 3475  
 3476  
 3477 022210 005713  
 3478 022212 005715  
 3479  
 3480 022214 005715  
 3481  
 3482  
 3483 022216 013701 177750  
 3484 022222 000240 25\$:  
 022224 000240  
 3485 022226 105037 177750  
 3486 022232 012737 001015 177746  
 3487 022240 032701 010000  
 3488 022244 001010  
 3489  
 3490 022246 010537 047334  
 3491  
 3492 022252 006237 047334  
 3493 022256 104406  
 022260 022256  
 3494  
 3495  
 3496  
 3497 022262 047334  
 3498

BIT #VSIU,CCR ;IS SET B BEING USED?  
 BNE 3\$ ;YES  
 BIS #FC,CCR ;NO: FLUSH CACHE FOR SET B  
 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
 BNE 200\$  
 MOV #60000,R5 ;:ADDRESS 60000 INTO R5  
 MOV #40000,R3 ;:ADDRESS 40000 INTO R3  
 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE  
 MOVB #HODO,CMR ;HODO ALLOWS VALID STORE SET B TO  
 ;BE WRITTEN TO CMR<12> ONLY DURING  
 ;THE DESTINATION MEMORY ACCESS.  
 TST (R3)  
 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS  
 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.  
 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>  
 ;FROM VALID STORE ADDRESS LOCATION  
 ;JUST WRITTEN INTO.  
 MOV CMR,R1 ;SAVE CMR DATA  
 25\$:  
 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP ;FOR LOOP ON ERROR  
 CLRB CMR ;DISABLE MAINT. MODE  
 MOV #OFF,CCR ;DISABLE CACHE  
 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.  
 BNE 9\$ ;PASS  
 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION  
 ;USED: CA<12:1>  
 ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
 ERROR ;ERROR  
 -----  
 .WORD -2 ;VALID BITS STORE TESTS - SET B  
 ;READING VALID STORE DATA SET B  
 ;THRU CMR<12> DID NOT RESULT IN 1.  
 CA121 ;PRINT VALID STORE ADDRESS LOCATION  
 ;USED: CA<12:1>.

CKKKABO 11-44 KK11B CACHE  
TEST # 133 - WRITE AND READ 1'S TO ALL LOW CACHE VALID

MACRO M1113 27-OCT-80 08:39 PAGE 112-1 SEQUENCE 140

K 11

3499	022264	000000		.WORD	0	
3500	022266	062705	000002	9\$: ADD	#2,R5	;NEXT VALID STORE LOCATION
3501	022272	062703	000002	ADD	#2,R3	
3502	022276	020527	070000	CMP	R5,#70000	;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
3503	022302	001334		BNE	1\$	;NO
3504	022304	000240		10\$: NOP		;END OF TEST
	022306	005237	001472	INC	\$TESTN	;INCREMENT TEST COUNTER

3510

```
.SBTTL TEST # 134 - WRITE AND READ 0'S TO ALL HIGH CACHE VALID
:*****
:*TEST 134 - WRITE AND READ 0'S TO ALL HIGH CACHE VALID
:*  WRITE AND READ 0'S TO ALL HIGH CACHE  VALID
:*  BIT STORE ADDRESS LOCATIONS- SET B
:*  (VALID STORE LOCATIONS 4000 TO 7777)
:*****
```

```
TST134:
3511 022312 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO LOW CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
                                .WORD 40$
                                .WORD 1$-40$+57764
                                .WORD 0
                                .WORD 25$-40$+57764
                                MOV #OFF,CCR
                                JSR R4,RELCTL
                                .WORD 10$+2
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE
```

```
3512 022340 032737 020000 177746          BIT #VSIU,CCR          ;IS SET B BEING USED?
3513 022346 001007          BNE 3$                ;YES
3514 022350 052737 000400 177746          BIS #FC,CCR           ;NO; FLUSH CACHE FOR SET B
3515 022356 032737 010000 177746 200$: BIT #VCIP,CCR         ;WAIT TILL FLUSH COMPLETE
3516 022364 001374          BNE 200$
3517 022366 012737 177777 177752 3$: MOV #-1,CHR           ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
3518                                ;REGISTERS, SINCE TDAR WILL BE USED
3519 022374 112737 000374 177751          MOVB #374,CMR+1
3520 022402 012705 070000          MOV #70000,R5         ;:ADDRESS 70000 INTO R5
3521 022406 012703 050000          MOV #50000,R3         ;:ADDRESS 50000 INTO R3
3522 022412 012737 000015 177746 1$: MOV #15,CCR           ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
3523 022420 112737 000003 177750          MOVB #HODO+TDAR,CMR  ;:HODO ALLOWS VALID STORE SET B TO
3524                                ;:BE WRITTEN TO CMR<12> ONLY DURING
3525                                ;:THE DESTINATION MEMORY ACCESS.
3526                                ;:TDAR WILL FORCE A 0 TO BE WRITTEN
3527                                ;:INTO VALID STORE WHEN A WRITE TO
3528                                ;:VALID STORE OCCURS
3529 022426 005713          TST (R3)
3530 022430 005715          TST (R5)              ;:WRITE A 0 INTO VALID STORE ADDRESS
3531                                ;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3532 022432 005715          TST (R5)              ;:WRITE VALID STORE DATA INTO CMR<12>
3533                                ;:FROM VALID STORE ADDRESS LOCATION
3534                                ;:JUST WRITTEN INTO.
3535 022434 013701 177750          MOV CMR,R1           ;:SAVE CMR DATA
3536 022440 000240          NOP                  ;:INSTRUCTION 'JMP 1$' PLACED HERE
3537 022442 000240          NOP                  ;:FOR LOOP ON ERROR
3538 022444 105037 177750          CLRB CMR            ;:DISABLE MAINT. MODE
3539 022450 012737 001015 177746          MOV #OFF,CCR         ;:DISABLE CACHE
3540 022456 032701 010000          BIT #VLD,R1         ;:CMR<12> SHOULD BE 0.
3541                                ;:PASS
3542 022464 010537 047334          MOV R5,CA121         ;:SAVE VALID STORE ADDRESS LOCATION
3543                                ;:USED: CA<12:1>
3544 022470 006237 047334          ASR CA121           ;:PREPARE CA121 FOR TYPEOUT
3545 022474 104406          ERROR              ;:ERROR
;-----
```

3546	022476	022474			.WORD	.-2		
3547								
3548								
3549	022500	047334			CA121			
3550								
3551	022502	000000			.WORD	0		
3552	022504	062705	000002	9\$:	ADD	#2,R5		
3553	022510	062703	000002		ADD	#2,R3		
3554	022514	020527	100000		CMP	R5,#100000		
3555	022520	001334			BNE	1\$		
3556	022522	000240		10\$:	NOP			
	022524	005237	001472		INC	\$TESTN		

```

:VALID BITS STORE TESTS - SET B
:READING VALID STORE DATA SET B
:THRU CMR<12> DID NOT RESULT IN 0.
:PRINT VALID STORE ADDRESS LOCATION
:USED: CA<12:1>.

:NEXT VALID STORE LOCATION

:HAVE ALL HIGH CACHE ADDRESS LOCATIONS BEEN DONE?
:NO
:END OF TEST
:INCREMENT TEST COUNTER
    
```



3562

.SBTTL TEST # 135 - WRITE AND READ 1'S TO ALL HIGH CACHE VALID  
 \*\*\*\*\*  
 \*TEST 135 - WRITE AND READ 1'S TO ALL HIGH CACHE VALID  
 \* WRITE AND READ 1'S TO ALL HIGH CACHE VALID  
 \* BIT STORE ADDRESS LOCATIONS- SET B  
 \* (VALID STORE LOCATIONS 4000 TO 7777)  
 \*\*\*\*\*

022530  
 3563 022530 000004  
 022532 022542  
 022534 060036  
 022536 000000  
 022540 060064  
 022542 012737 001015 177746  
 022550 004437 002342  
 022554 022726

TST135:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 ;TEST START LOCATION  
 .WORD 40\$ ;LOOP ON ERROR START LOCATION  
 .WORD 1\$-40\$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 0 ;LOOP ON ERROR END LOCATION  
 .WORD 25\$-40\$+57764 ;DISABLE CACHE  
 MOV #OFF,CCR ;LOCATE TEST CODE TO LOW CACHE SPACE  
 JSR R4,RELCTL ;ADDRESS OF START OF NEXT TEST  
 .WORD 10\$+2

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO LOW CACHE SPACE

3564 022556 032737 020000 177746  
 3565 022564 001007  
 3566 022566 052737 000400 177746  
 3567 022574 032737 010000 177746  
 3568 022602 001374  
 3569 022604 012705 070000  
 3570 022610 012703 050000  
 3571 022614 012737 000015 177746  
 3572 022622 112737 000002 177750  
 3573  
 3574  
 3575 022630 005713  
 3576 022632 005715  
 3577  
 3578 022634 005715  
 3579  
 3580  
 3581 022636 013701 177750  
 3582 022642 000240  
 022644 000240  
 3583 022646 105037 177750  
 3584 022652 012737 001015 177746  
 3585 022660 032701 010000  
 3586 022664 001010  
 3587 022666 010537 047334  
 3588  
 3589 022672 006237 047334  
 3590 022676 104406  
 022700 022676  
 3591  
 3592  
 3593  
 3594 022702 047334  
 3595  
 3596 022704 000000

BIT #VSIU,CCR ;IS SET B BEING USED?  
 BNE 3\$ ;YES  
 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B  
 200\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
 BNE 200\$  
 3\$: MOV #70000,R5 ;:ADDRESS 70000 INTO R5  
 MOV #50000,R3 ;:ADDRESS 50000 INTO R3  
 1\$: MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE  
 MOVB #HODO,CMR ;:HODO ALLOWS VALID STORE SET B TO  
 ;BE WRITTEN TO CMR<12> ONLY DURING  
 ;THE DESTINATION MEMORY ACCESS.  
 TST (R3)  
 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS  
 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.  
 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>  
 ;FROM VALID STORE ADDRESS LOCATION  
 ;JUST WRITTEN INTO.  
 MOV CMR,R1 ;SAVE CMR DATA  
 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 NOP ;FOR LOOP ON ERROR  
 CLRB CMR ;DISABLE MAINT. MODE  
 MOV #OFF,CCR ;DISABLE CACHE  
 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.  
 BNE 9\$ ;PASS  
 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION  
 ;USED: CA<12:1>  
 ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
 ERROR ;ERROR  
 ;-----  
 .WORD -2  
 ;VALID STORE BIT TEST- SET B  
 ;READING VALID STORE DATA SET B  
 ;THRU CMR<12> DID NOT RESULT IN 0.  
 CA121 ;PRINT VALID STORE ADDRESS LOCATION  
 ;USED: CA<12:1>.  
 .WORD 0

3597	022706	062705	000002	9\$:	ADD	#2,R5	;NEXT VALID STORE LOCATION
3598	022712	062703	000002		ADD	#2,R3	
3599	022716	020527	100000		CMP	R5,#100000	;HAVE ALL HIGH CACHE ADDRESS LOCATIONS BEEN DONE?
3600	022722	001334			BNE	1\$	;NO
3601	022724	000240		10\$:	NOP		;END OF TEST
	022726	005237	001472		INC	\$TESTN	;INCREMENT TEST COUNTER

TEST # 136 - CHK VALID DATA STORE ADRS LINES (CA(12:1)) SET B  
3605

.SBTTL TEST # 136 - CHK VALID DATA STORE ADRS LINES (CA(12:1)) SET B  
:\*\*\*\*\*  
:TEST 136 - CHK VALID DATA STORE ADRS LINES (CA(12:1)) SET B  
:\* VERIFY VALID DATA STORE ADDRESS LINES (CA<12:1>) - SET B  
:\*\*\*\*\*

022732  
3606 022732 000004  
  
022734 022744  
022736 070054  
022740 000000  
022742 070130  
022744 012737 001015 177746 40\$:  
022752 004437 002370  
022756 023174

TST136:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

3607 022760 032737 020000 177746  
3608 022766 001007  
3609 022770 052737 000400 177746  
3610 022776 032737 010000 177746 4\$:  
3611 023004 001374  
3612 023006 012737 000002 047344 3\$:  
3613 023014 012702 040000 2\$:  
3614 023020 012703 060000  
3615 023024 063702 047344  
3616 023030 063703 047344  
3617 023034 112737 000002 177750 1\$:  
3618  
3619  
3620 023042 012737 000015 177746  
3621 023050 152737 000001 177750  
3622  
3623  
3624 023056 005737 040000  
3625 023062 005737 060000  
3626 023066 142737 000001 177750  
3627  
3628  
3629 023074 005712  
3630 023076 005713  
3631  
3632 023100 005737 060000  
3633  
3634 023104 013701 177750  
3635 023110 000240 25\$:  
023112 000240  
3636 023114 105037 177750  
3637 023120 012737 001015 177746  
3638 023126 032701 010000  
3639 023132 001411  
3640  
3641 023134 013737 047344 047334  
3642 023142 006237 047334  
3643 023146 104406

BIT #VSIU,CCR ;IS SET B BEING USED?  
BNE 3\$ ;YES  
BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B  
BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
BNE 4\$  
MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN  
MOV #40000,R2 ;  
MOV #60000,R3 ;  
ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT  
ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT  
MOVB #HODO,CMR ;HODO ALLOWS VALID DATA STORE BITS TO BE  
;WRITTEN TO CMR<12> ONLY DURING THE  
;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.  
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE  
BISB #TDAR,CMR ;TDAR WILL FORCE A 0 TO BE WRITTEN  
;INTO VALID STORE WHEN A WRITE TO  
;VALID STORE OCCURS.  
TST 40000  
TST 60000 ;WRITE 0 INTO VALID STORE LOCATION 0000.  
BICB #TDAR,CMR ;CLEARING TDAR WILL ALLOW A 1 TO BE  
;WRITTEN INTO VALID STORE WHEN A WRITE  
;TO VALID STORE OCCURS.  
TST (R2)  
TST (R3) ;WRITE 1 INTO VALID STORE LOCATION  
;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.  
TST 60000 ;LOAD DATA FROM VALID DATA STORE LOCATION  
;0000 INTO CMR<12>.  
MOV CMR,R1 ;SAVE CMR DATA  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINT. MODE  
MOV #OFF,CCR ;DISABLE CACHE  
BIT #VLD,R1 ;CMR<12> SHOULD READ 0.  
BEQ 9\$ ;PASS  
;ROUTINE  
MOV FLTPAT,CA121 ;SAVE CA<12:1> USED  
ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
ERROR ;ERROR



3663

SBTTL TEST # 137 - ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS  
 \*\*\*\*\*  
 \*TEST 137 - ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS  
 \* VERIFY THAT ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS  
 \* WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.  
 \* (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 0000-3777)  
 \*\*\*\*\*

3664 023200 000004  
 023202 023212  
 023204 070074  
 023206 000000  
 023210 070130  
 023212 012737 001015 177746 40\$:  
 023220 004437 002370  
 023224 023452

TST137:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 ;TEST START LOCATION  
 .WORD 40\$ ;LOOP ON ERROR START LOCATION  
 .WORD 1\$-40\$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 .WORD 0 ;LOOP ON ERROR END LOCATION  
 .WORD 25\$-40\$+67764 ;DISABLE CACHE  
 MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE  
 JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST  
 .WORD 10\$+2

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

3665 023226 012705 060000  
 3666 023232 012703 040000  
 3667 023236 032737 020000 177746  
 3668 023244 001007  
 3669 023246 052737 000400 177746  
 3670 023254 032737 010000 177746 200\$:  
 3671 023262 001374  
 3672 023264 012737 000015 177746 3\$:  
 3673 023272 112737 000002 177750  
 3674  
 3675  
 3676 023300 005713 4\$:  
 3677 023302 005715  
 3678  
 3679 023304 062705 000002  
 3680 023310 062703 000002  
 3681 023314 020527 070000  
 3682 023320 001367  
 3683 023322 052737 000400 177746 1\$:  
 3684  
 3685 023330 032737 010000 177746 500\$:  
 3686 023336 001374  
 3687 023340 052737 000400 177746  
 3688 023346 032737 010000 177746 6\$:  
 3689 023354 001374  
 3690 023356 000240 25\$:  
 023360 000240  
 3691 023362 012705 060000 2\$:  
 3692 023366 005715  
 3693  
 3694  
 3695 023370 013701 177750  
 3696 023374 032701 010000  
 3697 023400 001416  
 3698 023402 105037 177750  
 3699 023406 012737 001015 177746

MOV #60000,R5 ;;ADDRESS 60000 INTO R5  
 MOV #40000,R3 ;;ADDRESS 40000 INTO R3  
 BIT #VSIU,CCR ;IS SET B BEING USED?  
 BNE 3\$ ;YES  
 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B  
 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
 BNE 200\$  
 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE  
 MOVB #HODO,CMR ;HODO ALLOWS VALID STORE SET B TO  
 ;BE WRITTEN TO CMR<12> ONLY DURING  
 ;THE DESTINATION MEMORY ACCESS.  
 TST (R3) 4\$:  
 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS  
 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.  
 ;NEXT VALID STORE LOCATION  
 ADD #2,R5  
 ADD #2,R3  
 CMP R5,#70000 ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?  
 BNE 4\$ ;NO  
 BIS #FC,CCR ;FLUSH CACHE TO SELECT SET A AND  
 ;INVALIDATE SET B  
 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE  
 BNE 500\$  
 BIS #FC,CCR ;FLUSH TO SELECT SET B AGAIN  
 BIT #VCIP,CCR ;WAIT  
 BNE 6\$  
 NOP 25\$:  
 NOP  
 MOV #60000,R5 ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 TST (R5) ;FOR LOOP ON ERROR  
 ;ADDRESS 60000 INTO R5  
 ;WRITE VALID STORE DATA INTO CMR<12>  
 ;FROM ADDRESS LOCATION SPECIFIED BY  
 ;R5'S BITS 12-1.  
 MOV CMR,R1 ;SAVE CMR DATA  
 BIT #VLD,R1 ;CMR<12> SHOULD BE 0  
 BEQ 9\$ ;PASS  
 CLRB CMR ;DISABLE MAINT MODE  
 MOV #OFF,CCR ;DISABLE CACHE

3700	023414	010537	047342		MOV	R5,CNT121		:SAVE VALID STORE FLUSH ADDRESS LOCATION
3701								:USED: CNT<12:1>
3702	023420	006237	047342		ASR	CNT121		:PREPARE CNT121 FOR TYPEOUT
3703	023424	104406			ERROR			:ERROR
								:-----
	023426	023424			.WORD	.-2		:FLUSH CACHE INVALID TEST-SET B
3704								:READING VALID STORE LOCATION FROM SET B THRU CMR<12>
3705								:DID NOT RESULT IN A ZERO,INDICATING THAT
3706								:THE CACHE FLUSH DID NOT INVALIDATE THIS
3707								:LOCATION.
3708								:PRINT VALID STORE FLUSH ADDRESS LOCATION
3709	023430	047342			CNT121			:IN ERROR: CNT<12:1>.
3710								
3711	023432	000000			.WORD	0		
3712	023434	000405			BR	10\$		:IF ERROR, END TEST
3713	023436	062705	000002	9\$:	ADD	#2,R5		:NEXT VALID STORE LOCATION
3714	023442	020527	070000		CMP	R5,#70000		:HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
3715	023446	001347			BNE	2\$		:NO
3716	023450	000240		10\$:	NOP			:END OF TEST
	023452	005237	001472		INC	\$TESTN		:INCREMENT TEST COUNTER

SBTTL TEST # 140 - ALL HI CACHE VALID STORE SET B ADDRESS LOCATIONS  
 \*\*\*\*\*  
 \*TEST 140 - ALL HI CACHE VALID STORE SET B ADDRESS LOCATIONS  
 \* VERIFY THAT ALL HI CACHE VALID STORE SET B ADDRESS LOCATIONS  
 \* WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.  
 \* (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 4000-7777)  
 \*\*\*\*\*

```

3723 023456 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO LOW CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
                                40$
                                .WORD 40$
                                .WORD 1$-40$+57764
                                .WORD 0
                                .WORD 25$-40$+57764
023460 023470          .WORD 40$
023462 060074          .WORD 1$-40$+57764
023464 000000          .WORD 0
023466 060130          .WORD 25$-40$+57764
023470 012737 001015 177746 40$: MOV #OFF,CCR
023476 004437 002342 JSR R4,RELCTL
023502 023730          .WORD 10$+2
  
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO LGW CACHE SPACE

```

3724 023504 012705 070000          MOV #70000,R5          ;:ADDRESS 70000 INTO R5
3725 023510 012703 050000          MOV #50000,R3          ;:ADDRESS 50000 INTO R3
3726 023514 032737 020000 177746          BIT #VSIU,CCR          ;:IS SET B BEING USED?
3727 023522 001007          BNE 3$                 ;:YES
3728 023524 052737 000400 177746          BIS #FC,CCR            ;:NO; FLUSH CACHE FOR SET B
3729 023532 032737 010000 177746 200$: BIT #VCIP,CCR          ;:WAIT TILL FLUSH COMPLETE
3730 023540 001374          BNE 200$
3731 023542 012737 000015 177746 3$: MOV #15,CCR            ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
3732 023550 112737 000002 177750          MOV #HODO,CMR          ;:HODO ALLOWS VALID STORE SET B TO
3733          ;:BE WRITTEN TO CMR<12> ONLY DURING
3734          ;:THE DESTINATION MEMORY ACCESS.
3735 023556 005713          4$: TST (R3)
3736 023560 005715          TST (R5)              ;:WRITE A 1 INTO VALID STORE ADDRESS
3737          ;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3738 023562 062705 000002          ADD #2,R5              ;:NEXT VALID STORE LOCATION
3739 023566 062703 000002          ADD #2,R3
3740 023572 020527 100000          CMP R5,#100000        ;:HAVE ALL HI CACHE LOCATIONS BEEN DONE?
3741 023576 001367          BNE 4$                 ;:NO
3742 023600 052737 000400 177746 1$: BIS #FC,CCR            ;:FLUSH CACHE TO SELECT SET A AND
3743          ;:INVALIDATE SET B
3744 023606 032737 010000 177746 500$: BIT #VCIP,CCR          ;:WAIT TILL FLUSH COMPLETE
3745 023614 001374          BNE 500$
3746 023616 052737 000400 177746          BIS #FC,CCR            ;:FLUSH TO SELECT SET B AGAIN
3747 023624 032737 010000 177746 6$: BIT #VCIP,CCR          ;:WAIT
3748 023632 001374          BNE 6$
3749 023634 000240          25$: NOP
                                ;:INSTRUCTION 'JMP 1$' PLACED HERE
                                ;:FOR LOOP ON ERROR
023636 000240          NOP
3750 023640 012705 070000          MOV #70000,R5          ;:ADDRESS 70000 INTO R5
3751 023644 005715          2$: TST (R5)          ;:WRITE VALID STORE DATA INTO CMR<12>
3752          ;:FROM ADDRESS LOCATION SPECIFIED BY
3753          ;:R5'S BITS 12-1.
3754 023646 013701 177750          MOV CMR,R1            ;:SAVE CMR DATA
3755 023652 032701 010000          BIT #VLD,R1          ;:CMR<12> SHOULD BE 0
3756 023656 001416          BEQ 9$                 ;:PASS
3757 023660 105037 177750          CLRB CMR              ;:DISABLE MAINT MODE
3758 023664 012737 001015 177746          MOV #OFF,CCR          ;:DISABLE CACHE
  
```

```

3759 023672 010537 047342      MOV      R5,CNT121      ;SAVE VALID STORE FLUSH ADDRESS LOCATION
3760                               ;USED: CNT<12:1>
3761 023676 006237 047342      ASR      CNT121        ;PREPARE CNT121 FOR TYPEOUT
3762 023702 104406              ERROR              ;ERROR
                               ;-----
      023704 023702              .WORD     .-2
3763                               ;FLUSH CACHE INVALID TEST-SET B
3764                               ;READING VALID STORE LOCATION FROM SET B THRU CMR<12>
3765                               ;DID NOT RESULT IN A ZERO,INDICATING THAT
3766                               ;THE CACHE FLUSH DID NOT INVALIDATE THIS
3767                               ; LOCATION.
3768 023706 047342              CNT121            ;PRINT VALID STORE FLUSH ADDRESS LOCATION
3769                               ;IN ERROR: CNT<12:1>.
3770 023710 000000              .WORD     0
3771 023712 000405              BR        10$      ;IF ERROR,END TEST
3772 023714 062705 000002      9$: ADD     #2,R5    ;NEXT VALID STORE LOCATION
3773 023720 020527 100000      CMP      R5,#100000 ;HAVE ALL HI CACHE ADDRESS LOCATIONS BEEN DONE?
3774 023724 001347              BNE      2$        ;NO
3775 023726 000240              10$: NOP
      023730 005237 001472      INC      $TESTN    ;INCREMENT TEST COUNTER
  
```



3781

```
.SBTTL TEST # 141 - TEST UPDATE TO CACHE DATA ON HIT/MISS
:*****
:TEST 141 - TEST UPDATE TO CACHE DATA ON HIT/MISS
:*
:* VERIFY THE FOLLOWING:
:* 1. NO UPDATE OCCURS TO CACHE DATA STORE DUE TO A WRITE MISS
:* 2. UPDATE DOES OCCUR TO CACHE DATA STORE DUE TO A WRITE HIT
:*****
```

```
TST141:
3782 023734 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
023736 023746 .WORD 40$ ;TEST START LOCATION
023740 070000 .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
023742 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
023744 070064 .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
023746 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
023754 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
023760 024154 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

```
3783 023762 005037 060000 1$: CLR 60000 ;0'S TO MAIN MEMORY LOCATION
3784 023766 005037 000000 CLR 0 ;CLEAR LOCATION 0
3785 023772 012700 177777 MOV #-1,R0 ;ALL 1'S TO R0
3786 023776 012701 060000 MOV #60000,R1 ;ADDRESS 60000 TO R1
3787 024002 112737 000002 177750 MOVB #HODO,CMR ;ALLOWS CACHE UPDATES & DATA STORE BITS TO BE
3788 ;WRITTEN TO CDR<15:0> ONLY DURING THE
3789 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3790 024010 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3791 024016 005737 060000 TST 60000 ;
3792 024022 005737 000000 TST 0 ;READ UPDATE; ALL 0'S TO DATA STORE
3793 ;LOCATION 0000 FROM MAIN MEMORY
3794 ;LOC. 0
3795 024026 010011 MOV R0,(R1) ;WRITE MISS:NO UPDATE SHOULD OCCUR
3796 ;TO DATA STORE LOCATION 0000
3797 024030 005711 TST (R1) ;READ MISS;LOAD DATA STORE BITS RESULTING
3798 ;FROM PREVIOUS WRITE MISS INTO CDR<15:0>
3799 024032 013702 177754 MOV CDR,R2 ;SAVE CDR CONTENTS
3800 024036 010011 MOV R0,(R1) ;WRITE HIT;
3801 ;THIS WRITE HIT SHOULD UPDATE DATA
3802 ;STORE LOCATION 0000.
3803 024040 005711 TST (R1) ;READ HIT;LOAD DATA STORE BITS RESULTING
3804 ;FROM PREVIOUS WRITE HIT INTO CDR<15:0>
3805 024042 013703 177754 MOV CDR,R3 ;SAVE CDR CONTENTS
3806 024046 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
024050 000240 NOP ;FOR LOOP ON ERROR
3807 024052 105037 177750 CLR RB CMR ;DISABLE MAINTENANCE
3808 024056 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3809 024064 012737 000002 000000 MOV #2,0 ;RESTORE VECTOR
3810 024072 005702 TST R2 ;CHECK FOR ALL 0'S
3811 024074 001411 BEQ 8$ ;PASS
3812 024076 005037 047332 CLR EXDAT6 ;SPECIFY EXPECTED DATA
3813 024102 010237 047330 MOV R2,CDR150 ;GET RECEIVED DATA FROM R2
3814 024106 104406 ERROR ;ERROR
;-----
024110 024106 .WORD -2
3815 ;WRITE CONTROL LOGIC TEST
```



.SBTTL TEST # 142 - TEST WRITE CONTROL LOGIC INHIBIT MODE  
\*\*\*\*\*  
\*TEST 142 - TEST WRITE CONTROL LOGIC INHIBIT MODE  
\* VERIFY THAT THE WRITE CONTROL LOGIC WILL BE INHIBITED FROM UPDATING  
\* TAG STORE DUE TO A READ HIT.  
\*PROCEDURE: CREATE READ HIT TO LOW CACHE WITH FMLO ENABLED. FMLO WILL  
\* INHIBIT CPU RESTART SIGNAL SO THAT A POTENTIAL WRITE SIGNAL COULD  
\* CONTROL LOGIC SHOULD BE INHIBITED DUE TO READ HIT.  
\*\*\*\*\*

024160  
3843 024160 000004

024162 024172  
024164 070000  
024166 000000  
024170 070064  
024172 012737 001015 177746  
024200 004437 002370  
024204 024364

TST142:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

3844 024206 012737 177777 177752 1\$:  
3845 024214 112737 000002 177750  
3846  
3847  
3848  
3849 024222 012737 000015 177746  
3850 024230 005737 040000  
3851 024234 005737 000000  
3852 024240 052737 000001 177750  
3853  
3854  
3855 024246 005737 000000  
3856  
3857  
3858 024252 005737 000000  
3859  
3860 024256 013737 177752 047322  
3861 024264 012737 001015 177746  
3862 024272 000240 25\$:  
024274 000240  
3863 024276 105037 177750  
3864 024302 042737 000177 047322  
3865 024310 005737 047322  
3866 024314 001422  
3867 024316 012737 000007 002062  
3868 024324 006237 047322 2\$:  
3869 024330 042737 100000 047322  
3870 024336 005337 002062  
3871 024342 001370  
3872 024344 005037 047350  
3873 024350 104406

MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>  
MOVB #HODO,CMR ;HODO ALLOWS CACHE TAG FIELD BITS TO BE  
;WRITTEN TO CHR<15:07> ONLY DURING  
;THE DESTINATION MEMORY ACCESS  
;OF AN INSTRUCTION  
;NO UCB SO AS TO WRITE ENABLE  
MOV #15,CCR  
TST 40000  
TST 0 ;READ UPDATE;LOAD TAG STORE WITH ALL 0'S  
BIS #TDAR,CMR ;TDAR WILL ALLOW TAG STORE TO BE  
;WRITTEN WITH CONTENTS OF AMR<8:0>  
;IF AN UPDATE OCCURS.  
TST 0 ;READ HIT; WRITE CONTROL LOGIC SHOULD  
;BE INHIBITED FROM ISSUING A WRITE  
;SIGNAL  
TST 0 ;WRITE TAG FIELD DATA FROM TAG STORE  
;LOCATION 0000 INTO CHR.  
MOV CHR,CHR157 ;SAVE CHR DATA  
MOV #OFF,CCR ;DISABLE CACHE  
25\$:  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINTENANCE MODE  
BIC #177,CHR157 ;INTERESTED IN 15:07  
TST CHR157 ;BITS 15:07 SHOULD BE ALL 0'S  
BEQ 10\$ ;PASS  
MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT  
2\$:  
ASR CHR157  
BIC #100000,CHR157  
DEC LOOP  
BNE 2\$  
CLR EXDAT3 ;INDICATE EXPECTED DATA  
ERROR ;ERROR  
;-----

024352 024350  
3874

.WORD -2  
;WRITE CONTROL LOGIC TESTS

3875  
3876  
3877  
3878  
3879  
3880 024354 047350  
3881 024356 047322  
3882 024360 000000  
3883 024362 000240  
          024364 005237 001472

EXDAT3  
CHR157  
.WORD 0  
10\$: NOP  
INC \$TESTN

:READING TAG STORE DATA THRU CHR<15:07>  
:DID NOT RESULT IN ALL 0'S.  
:THIS SUGGESTS THAT AN UPDATE OCCURED  
:AND WRITE CONTROL LOGIC WAS NOT  
:INHIBITED DUE TO READ HIT.  
:PRINT CHR<15:07> EXPECTED  
:PRINT CHR<15:07> RECEIVED  
  
:END OF TEST  
:INCREMENT TEST COUNTER

```
.SBTTL TEST # 143 - WRITE CONTROL LOGIC INHIBIT TEST
:*****
:*TEST 143 - WRITE CONTROL LOGIC INHIBIT TEST
:*  VERIFY THAT WRITE CONTROL LOGIC WILL INHIBIT A READ UPDATE
:*  TO CACHE TAG STORE DUE TO AN ACCESS TO I/O PAGE.
:*****
```

```
TST143:
3889 024370 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      024372 024402      .WORD 40$      ;ERROR/LOOP ON TEST
      024374 060000      .WORD 1$-40$+57764 ;TEST START LOCATION
      024376 000000      .WORD 0         ;LOOP ON ERROR START LOCATION
      024400 060042      .WORD 25$-40$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR - THIS TEST
      024402 012737 001015 177746 40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
      024410 004437 002342      JSR R4,RELCTL ;DISABLE CACHE
      024414 024564      .WORD 10$+2    ;LOCATE TEST CODE TO LOW CACHE SPACE
                               ;ADDRESS OF START OF NEXT TEST
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE
```

```
3890 024416 112737 000002 177750 1$:  MOVB #HODO,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
3891                                     ;WRITTEN TO CHR<15:07> ONLY DURING
3892                                     ;THE DESTINATION MEMORY ACCESS
3893                                     ;OF AN INSTRUCTION
3894 024424 012737 000015 177746      MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3895 024432 005737 057744      TST 57744
3896 024436 005737 077744      TST 77744 ;READ UPDATE;LOAD BIT PATTERN
3897                                     ;000000011 INTO TAG STORE LOCATION
3898                                     ;7762
3899 024442 005737 177744      TST 177744 ;ACCESS I/O PAGE BY READING CCR REGISTER.
3900                                     ;THE CACHE COULD DO AN UPDATE TO
3901                                     ;TAG STORE LOCATION 7762 BUT THE ACCESS
3902                                     ;TO I/O PAGE WILL INHIBIT WRITE CONTROL
3903                                     ;LOGIC
3904 024446 005737 057744      TST 57744 ;WRITE TAG STORE DATA FROM LOCATION
3905                                     ;7762 INTO CHR<15:07>.
3906 024452 013737 177752 047322      MOV CHR,CHR157 ;SAVE CHR DATA
3907 024460 000240 25$:  NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      024462 000240      NOP ;FOR LOOP ON ERROR
3908 024464 105037 177750      CLRB CMR ;DISABLE MAINTENANCE MODE
3909 024470 012737 001015 177746      MOV #OFF,CCR ;DISABLE CACHE
3910 024476 042737 000177 047322      BIC #177,CHR157 ;PREPARE CHR157 FOR ERROR CHECK
3911 024504 022737 000600 047322      CMP #600,CHR157 ;BITS 15:07 SHOULD BE BIT PATTERN 000000011
3912 024512 001423      BEQ 10$ ;PASS
3913 024514 012737 000007 002062      MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
3914 024522 006237 047322 2$:  ASR CHR157
3915 024526 042737 100000 047322      BIC #100000,CHR157
3916 024534 005337 002062      DEC LOOP
3917 024540 001370      BNE 2$
3918 024542 012737 000003 047350      MOV #3,EXDAT3 ;INDICATE EXPECTED DATA
3919 024550 104406      ERROR ;ERROR
      024552 024550      .WORD -2 ;-----
3920                                     ;WRITE CONTROL LOGIC TESTS
3921                                     ;READING TAGD<21:13> THRU CHR<15:07>
3922                                     ;DID NOT RESULT IN BIT PATTERN 000000011.
3923 024554 047350      EXDAT3 ;PRINT CHR 15:07 EXPECTED
```

3924 024556 047322  
3925 024560 000000  
3926 024562 000240  
          024564 005237 001472

10\$:

CHR157  
.WORD 0  
NOP  
INC \$TESTN

;PRINT CHR <15:07> RECEIVED  
;END OF TEST  
;INCREMENT TEST COUNTER

TEST # 144 - WRITE CONTROL AND VALID STORE LOGIC TEST

3940

.SBTTL TEST # 144 - WRITE CONTROL AND VALID STORE LOGIC TEST

\*\*\*\*\*

\*TEST 144 - WRITE CONTROL AND VALID STORE LOGIC TEST

\* THIS TEST VERIFIES THE AREA OF WRITE CONTROL LOGIC AND VALID STORE LOGIC THAT IS CONCERNED WITH BYPASS OPERATIONS. A WIRE STRAP IS USED TO ALLOW OR INHIBIT INVALIDATION OF VALID STORE DURING READ BYPASS CONDITIONS. UNLESS SWITCH REGISTER 08 IS IMPLEMENTED, THIS TEST ASSUMES THAT STRAP W1 IS IN PLACE.

\*PROCEDURE: IF SWR 08 IS NO' IMPLEMENTED, W1 IS ASSUMED IN PLACE. NO INVALIDATION OF VALID STORE SET A SHOULD OCCUR DUE TO READ MISS/BYPASS AND READ HIT/BYPASS CONDITIONS. 2. IF SWR 08 IS IMPLEMENTED, STRAP W2 IS ASSUMED IN PLACE. NO INVALIDATION SHOULD OCCUR DUE TO READ MISS/BYPASS, BUT INVALIDATION SHOULD OCCUR DUE TO READ HIT/BYPASS CONDITION.

\*\*\*\*\*

TST144:

3941	024570	000004				SCPCND		;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	024572	024602				.WORD	40\$	;ERROR/LOOP ON TEST
	024574	070026				.WORD	1\$-40\$+67764	;TEST START LOCATION
	024576	000000				.WORD	0	;LOOP ON ERROR START LOCATION
	024600	070112				.WORD	25\$-40\$+67764	;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	024602	012737	001015	177746	40\$:	MOV	#OFF,CCR	;LOOP ON ERROR END LOCATION
	024610	004437	002370			JSR	R4,RELCTH	;DISABLE CACHE
	024614	025036				.WORD	10\$+2	;LOCATE TEST CODE TO HIGH CACHE SPACE
								;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

3942	024616	032737	020000	177746		BIT	#VSIU,CCR	;IS SET A BEING USED?
3943	024624	001407				BEQ	1\$	;YES
3944	024626	052737	000400	177746		BIS	#FC,CCR	;NO,FLUSH CACHE FOR SET A
3945	024634	032737	010000	177746	200\$:	BIT	#VCIP,CCR	;WAIT TILL FLUSH COMPLETE
3946	024642	001374				BNE	200\$	
3947	024644	012737	000015	177746	1\$:	MOV	#15,CCR	;NO UCB SO AS TO ENABLE CACHE STORES
3948	024652	112737	000002	177750		MOVB	#HODO,CMR	;HODO ALLOWS READ UPDATES,AND VALID
3949								;STORE DATA TO BE WRITTEN TO CMR <12>
3950								;ONLY DURING THE DESTINATION ACCESS
3951								;OF AN INSTRUCTION.
3952	024660	005737	040000			TST	40000	
3953	024664	005737	060000			TST	60000	;READ UPDATE TO CACHE LOCATION 0000.
3954								;WRITE 1 INTO VALID STORE LOCATION 0000.
3955	024670	052737	001000	177746		BIS	#UCB,CCR	;BYPASS MODE
3956	024676	005737	040000			TST	40000	;READ MISS/BYPASS;
3957	024702	005737	060000			TST	60000	;LOAD VALID STORE LOCATION 0000 DATA
3958								;RESULTING FROM PREVIOUS READ MISS/BYPASS
3959								;INTO CMR<12>.
3960								;THIS IS ALSO A READ HIT/BYPASS CONDITION.
3961	024706	013700	177750			MOV	CMR,R0	;SAVE CMR CONTENTS
3962	024712	005737	040000			TST	40000	;LOAD VALID STORE LOCATION 0000
3963								;DATA RESULTING FROM PREVIOUS READ HIT
3964								; /BYPASS INTO CMR<12>.
3965	024716	013701	177750			MOV	CMR,R1	;SAVE CMR CONTENTS
3966	024722	012737	001015	177746		MOV	#OFF,CCR	;DISABLE CACHE
3967	024730	000240			25\$:	NOP		;INSTRUCTION 'JMP 1\$' PLACED HERE
	024732	000240				NOP		;FOR LOOP ON ERROR
3968	024734	105037	177750			CLRB	CMR	;DISABLE MAINTENANCE MODE

```

3969 024740 012702 002074      MOV      #SWR,R2      ;
3970 024744 032732 000400      BIT      #BIT08,@(R2)+ ;IS STRAP W2 IN PLACE
3971 024750 001015              BNE      7$          ;YES
3972 024752 032700 010000      BIT      #VLD,R0     ;NO; W1 IS ASSUMED
3973 024756 001003              BNE      6$          ;PASS ;VALID DATA IS A 1
3974 024760 104406              ERROR     ;ERROR
;-----
          024762 024760          .WORD    .-2
3975                                ;WRITE CONTROL LOGIC TESTS
3976                                ;STRAP W1 IS ASSUMED IN PLACE.
3977                                ;READ MISS/BYPASS CAUSED INVALIDATION
3978                                ;OF LOCATION 0000.
3979 024764 000000              .WORD    0
3980 024766 032701 010000      6$:     BIT      #VLD,R1 ;TEST VALID DATA HELD IN R1
3981 024772 001020              BNE      10$         ;PASS; STILL A 1
3982 024774 104406              ERROR     ;ERROR
;-----
          024776 024774          .WORD    .-2
3983                                ;WRITE CONTROL LOGIC TESTS
3984                                ;STRAP W1 IS ASSUMED IN PLACE.
3985                                ;READ MISS/HIT CAUSED INVALIDATION
3986                                ;OF LOCATION 0000.
3987 025000 000000              .WORD    0
3988 025002 000414              BR       10$
3989 025004 032700 010000      7$:     BIT      #VLD,R0 ; W2 IS ASSUMED IN PLACE
3990 025010 001003              BNE      8$          ;PASS ;VALID DATA IS A 1
3991 025012 104406              ERROR     ;ERROR
;-----
          025014 025012          .WORD    .-2
3992                                ;WRITE CONTROL LOGIC TESTS
3993                                ;STRAP W2 IS ASSUMED IN PLACE.
3994                                ;READ MISS/BYPASS CAUSED INVALIDATION
3995                                ;OF LOCATION 0000.
3996 025016 000000              .WORD    0
3997 025020 032701 010000      8$:     BIT      #VLD,R1 ;TEST VALID DATA HELD IN R1
3998 025024 001403              BEQ     10$         ;PASS; VALID DATA IS A 0
3999 025026 104406              ERROR     ;ERROR
;-----
          025030 025026          .WORD    .-2
4000                                ;WRITE CONTROL LOGIC TESTS
4001                                ;STRAP W2 IS ASSUMED IN PLACE.
4002                                ;READ MISS/HIT DID NOT CAUSE INVALIDATION
4003                                ;OF LOCATION 0000.
4004 025032 000000              .WORD    0
4005 025034 000240              10$:    NOP
          025036 005237 001472      INC     $TESTN      ;END OF TEST
;INCREMENT TEST COUNTER

```



4006

.SBTTL TEST # 145 - WRITE HIT IN BYPASS MODE INVALIDATES CACHE LOCATION

\*\*\*\*\*  
 \*TEST 145 - WRITE HIT IN BYPASS MODE INVALIDATES CACHE LOCATION  
 \*\*\*\*\*

TST145:

4007 025042 000004

SCPCND

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 :ERROR/LOOP ON TEST

025044 025054  
 025046 070026  
 025050 000000  
 025052 070102  
 025054 012737 001015 177746 40\$:  
 025062 004437 002370  
 025066 025220

.WORD 40\$  
 .WORD 1\$-40\$+67764  
 .WORD 0  
 .WORD 25\$-40\$+67764  
 MOV #OFF,CCR  
 JSR R4,RELCTH  
 .WORD 10\$+2

:TEST START LOCATION  
 :LOOP ON ERROR START LOCATION  
 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
 :LOOP ON ERROR END LOCATION  
 :DISABLE CACHE  
 :LOCATE TEST CODE TO HIGH CACHE SPACE  
 :ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
 :ARE RELOCATED TO HI CACHE SPACE

4008 025070 032737 020000 177746

BIT #VSIU,CCR

:IS SET A BEING USED?

4009 025076 001407

BEQ 1\$

:YES

4010 025100 052737 000400 177746

BIS #FC,CCR

:NO,FLUSH CACHE FOR SET A

4011 025106 032737 010000 177746

200\$:

BIT #VCIP,CCR

:WAIT TILL FLUSH COMPLETE

4012 025114 001374

BNE 200\$

4013 025116 012700 000002

1\$:

MOV #2,R0

:DATA TO R0

4014 025122 005001

CLR R1

:ADDRESS 0 TO R1

4015 025124 012737 000015 177746

MOV #15,CCR

:NO UCB SO AS TO ENABLE CACHE STORES

4016 025132 112737 000002 177750

MOV #HODO,CMR

:HODO ALLOWS READ UPDATES,AND VALID

4017

:STORE DATA TO BE WRITTEN TO CMR <12>

4018

:ONLY DURING THE DESTINATION ACCESS

4019

:OF AN INSTRUCTION.

4020 025140 005737 040000

TST 40000

4021 025144 005711

TST (R1)

:READ UPDATE TO CACHE LOCATION 0000.

4022

:WRITE 1 INTO VALID STORE LOCATION 0000.

4023 025146 052737 001000 177746

BIS #UCB,CCR

:BYPASS MODE

4024 025154 010011

MOV R0,(R1)

:WRITE HIT BYPASS TO LOC. 0 SHOULD INVALIDATE

4025 025156 005711

TST (R1)

:LOAD VALID STORE LOCATION 0000 DATA

4026

:RESULTING FROM PREVIOUS WRITE HIT/BYPASS INTO CMR<12>.

4027 025160 013702 177750

MOV CMR,R2

:SAVE CMR CONTENTS

4028 025164 012737 001015 177746

MOV #OFF,CCR

:DISABLE CACHE

4029 025172 000240

25\$:

NOP

:INSTRUCTION 'JMP 1\$' PLACED HERE

4030 025176 105037 177750

NOP

:FOR LOOP ON ERROR

4031 025202 032702 010000

CLRB CMR

:DISABLE MAINTENANCE MODE

4032 025206 001403

BIT #VLD,R2

:CHECK FOR 0

4033 025210 104406

BEQ 10\$

:PASS

4034

:ERROR

4035

:-----

4036

.WORD -2

:WRITE CONTROL LOGIC TESTS

4037 025214 000000

.WORD 0

:WRITE HIT /BYPASS DID NOT INVALIDATE

4038 025216 000240

10\$:

NOP

:CACHE VALID STORE LOCATION

025220 005237 001472

INC \$TESTN

:END OF TEST  
 :INCREMENT TEST COUNTER

```

.SBTTL TEST # 146 - VERIFY CACHE DATA STORE RAM MEMORY IC'S
*****
*TEST 146 - VERIFY CACHE DATA STORE RAM MEMORY IC'S
*VERIFY CACHE DATA STORE RAM MEMORY IC'S BY PERFORMING A
* MARCH PATTERN TEST TO LOW CACHE AREA OF DATA STORE(LOC. 0000-3777)
* PROCEDURE: 1. WRITE ALL 0'S TO ALL LO CACHE DATA STORE
* RAMS CORRESPONDING TO LOCATIONS 0000-3777
* 2. READ 0'S FROM ALL LO CACHE RAMS CORRESPONDING
* TO LOCATION 0000
* 3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
* 0000.
* 4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
* 0000.
* 5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
* AND UNTIL LOC. 3777 IS REACHED.
*****
  
```

```

4054 025224 000004          TST146:          SPCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          025226 025236          .WORD 40$          ;TEST START LOCATION
          025230 070000          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
          025232 000000          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          025234 070100          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
          025236 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
          025244 004437 002370          JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
          025250 025540          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST

          ;THE FOLLOWING LOCATIONS INCLUDING 10$
          ;ARE RELOCATED TO HI CACHE SPACE

4055 025252 012700 060000 1$: MOV #60000,R0 ;ADDRESS LOC. 60000 TO R0
4056 025256 005020 5$: CLR (R0)+ ;CLEAR ALL LOW CACHE MAIN MEMORY
4057 025260 020027 070000 CMP R0,#70000 ;DONE?
4058 025264 001374 BNE 5$ ;NO
4059 025266 012700 060000 MOV #60000,R0 ;ADDR. LOC. 60000 TO R0
4060 025272 012701 040000 MOV #40000,R1 ;ADDR. LOC. 40000 TO R1
4061 025276 012702 177777 MOV #-1,R2 ;R2 CONTAINS ALL 1'S
4062 025302 112737 000002 177750 MOVVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & DATA STORE BITS
4063 ;TO BE WRITTEN TO CDR<15:0> ONLY DURING
4064 ;THE DESTINATION MEMORY ACCESS OF AN
4065 ;INSTRUCTION
4066 025310 012737 000015 177746 MOV #15,CCR ;WRITE ENABLE CACHE DATA STORES
4067 025316 005721 6$: TST (R1)+ ;UPDATE ALL LOW CACHE DATA STORE WITH 0'S
4068 025320 005720 TST (R0)+ ;
4069 025322 020027 070000 CMP R0,#70000 ;DONE?
4070 025326 001373 BNE 6$ ;NO
4071 025330 012700 060000 MOV #60000,R0 ;ADDR. 60000 TO R0
4072 025334 005710 7$: TST (R0) ;READ HIT TO CACHE DATA STORE LOCATION
4073 ;SPECIFIED BY R0.CLOCK DATA STORE
4074 ;BITS INTO CDR<15:0>.SHOULD BE ALL 0'S.
4075 025336 013705 177754 MOV CDR,R5 ;SAVE CDR CONTENTS
4076 025342 010210 MOV R2,(R0) ;WRITE HIT CACUSES UPDATE TO CACHE DATA
4077 ;STORE LOCATION.WRITE ALL 1'S.
4078 025344 005710 TST (R0) ;READ HIT.CLOCK DATA STORE BITS TO
4079 ;CDR <15:0>.SHOULD BE ALL 1'S.
4080 025346 013703 177754 MOV CDR,R3 ;SAVE CDR CONTENTS
4081 025352 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
  
```

4082	025354	000240				NOP			:FOR LOOP ON ERROR
4083	025356	005705				TST	R5		:SHOULD BE ALL 0'S
4084	025360	001424				BEQ	8\$		:PASS
4085	025362	012737	001015	177746		MOV	#OFF,CCR		:DISABLE CACHE
4086	025370	105037	177750			CLRB	CMR		:CLEAR MAINT. MODE
4087	025374	005037	047332			CLR	EXDAT6		:SPECIFY EXPECTED CACHE DATA STORE DATA
4088	025400	010537	047330			MOV	R5,CDR150		:SPECIFY CACHE DATA STORE DATA READ THRU CDR<15:0>
4089	025404	010037	047334			MOV	R0,CA121		:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4090	025410	006237	047334			ASR	CA121		:ERROR
	025414	104406				ERROR			:-----
4091	025416	025414				.WORD	.-2		:DATA STORE MARCH PATTERN TEST
4092									:READING CACHE DATA STORE DATA
4093									:THRU CDR<15:0> DID NOT READ ALL 0'S.
4094									:THIS SUGGESTS THAT A RAM LOCATION
4095									:SPECIFIED BY CA121 WAS OVERWRITTEN
4096									:WITH A 1 WHEN WRITING A 1 TO ANOTHER
4097									:LOCATION.ANY BIT IN CDR150 DATA
4098									:THAT IS A 1 MAY POINT TO A BAD
4099									:CACHE DATA STORE RAM.
4100	025420	047332				EXDAT6			: EXPECTED CACHE DATA STORE DATA
4101	025422	047330				CDR150			: CACHE DATA STORE DATA READ
4102									:THRU CDR<15:0>
4103	025424	047334				CA121			:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4104	025426	000000				.WORD	0		
4105	025430	000435				BR	3\$		:END THE TEST
4106	025432	022703	177777		8\$:	CMP	#-1,R3		:SHOULD BE ALL 1'S
4107	025436	001425				BEQ	9\$		:PASS
4108	025440	012737	001015	177746		MOV	#OFF,CCR		:DISABLE CACHE
4109	025446	105037	177750			CLRB	CMR		:CLEAR MAINT. MODE
4110	025452	012737	177777	047332		MOV	#-1,EXDAT6		:SPECIFY EXPECTED CACHE DATA STORE DATA
4111	025460	010337	047330			MOV	R3,CDR150		:SPECIFY CACHE DATA STORE DATA READ
4112									:THRU CDR<15:0>
4113	025464	010037	047334			MOV	R0,CA121		:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4114	025470	006237	047334			ASR	CA121		:ERROR
4115	025474	104406				ERROR			:-----
	025476	025474				.WORD	.-2		:DATA STORE MARCH PATTERN TEST
4116									:READING CACHE DATA STORE DATA
4117									:THRU CDR<15:0> DID NOT READ ALL 1'S.
4118									:ANY BIT IN CDR150 DATA
4119									:THAT IS A 0 MAY POINT TO A BAD
4120									:CACHE DATA STORE RAM.
4121									
4122									
4123	025500	047332				EXDAT6			: EXPECTED CACHE DATA STORE DATA
4124	025502	047330				CDR150			: CACHE DATA STORE DATA READ
4125									:THRU CDR<15:0>
4126	025504	047334				CA121			:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4127	025506	000000				.WORD	0		
4128	025510	000405				BR	3\$		:END TEST
4129	025512	062700	000002		9\$:	ADD	#2,R0		:NEXT LOCATION
4130	025516	022700	070000			CMP	#70000,R0		:HAS ALL LO CACHE BEEN DONE?
4131	025522	001304				BNE	7\$		:NO,CONTINUE
4132	025524	012737	001015	177746	3\$:	MOV	#OFF,CCR		:DISABLE CACHE
4133	025532	105037	177750			CLRB	CMR		:DISABLE MAINT. MODE

CKKKABO 11-44 KK11B CACHE MACRO M1113 27-OCT-80 08:39 PAGE 123-2 SEQUENCE 162  
TEST # 146 - VERIFY CACHE DATA STORE RAM MEMORY IC'S  
4134 025536 000240 10\$: NOP :END OF TEST  
025540 005237 001472 INC \$TESTN ;INCREMENT TEST COUNTER

```

.SBTTL TEST # 147 - VERIFY CACHE DATA STORE RAM MEMORY IC'S
*****
*TEST 147 - VERIFY CACHE DATA STORE RAM MEMORY IC'S
*VERIFY CACHE DATA STORE RAM MEMORY IC'S BY PERFORMING A
*MARCH PATTERN TEST TO HIGH CACHE AREA OF DATA STORE(LOC. 4000-7777)
*PROCEDURE: 1. WRITE ALL 0'S TO ALL HI CACHE DATA STORE
*RAMS CORRESPONDING TO LOCATIONS 4000-7777
*2. READ 0'S FROM ALL HI CACHE RAMS CORRESPONDING
*TO LOCATION 4000
*3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
*4000.
*4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
*4000.
*5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*AND UNTIL LOC. 7777 IS REACHED.
*****
    
```

```

TST147:
4150 025544 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          .WORD 40$          ;TEST START LOCATION
          .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
          .WORD 40$:        ;DISABLE CACHE
          .WORD 10$+2       ;LOCATE TEST CODE TO LOW CACHE SPACE
          ;ADDRESS OF START OF NEXT TEST
    
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO LOW CACHE SPACE

```

4151 025572 012700 070000 1$: MOV #70000,R0 ;ADDRESS LOC. 70000 TO R0
4152 025576 005020 5$: CLR (R0)+ ;CLEAR ALL HIGH CACHE MAIN MEMORY
4153 025600 020027 100000 CMP R0,#100000 ;DONE?
4154 025604 001374 BNE 5$ ;NO
4155 025606 012700 070000 MOV #70000,R0 ;ADDR. LOC. 70000 TO R0
4156 025612 012701 050000 MOV #50000,R1 ;ADDR. LOC. 50000 TO R1
4157 025616 012702 177777 MOV #-1,R2 ;R2 CONTAINS ALL 1'S
4158 025622 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & DATA STORE BITS
4159 ;TO BE WRITTEN TO CDR<15:0> ONLY DURING
4160 ;THE DESTINATION MEMORY ACCESS OF AN
4161 ;INSTRUCTION
4162 025630 012737 000015 177746 MOV #15,CCR ;WRITE ENABLE CACHE DATA STORES
4163 025636 005721 6$: TST (R1)+ ;UPDATE ALL HIGH CACHE DATA STORE WITH 0'S
4164 025640 005720 TST (R0)+
4165 025642 020027 100000 CMP R0,#100000 ;DONE?
4166 025646 001373 BNE 6$ ;NO
4167 025650 012700 070000 MOV #70000,R0 ;ADDR. 70000 TO R0
4168 025654 005710 7$: TST (R0) ;READ HIT TO CACHE DATA STORE LOCATION
4169 ;SPECIFIED BY R0.CLOCK DATA STORE
4170 ;BITS INTO CDR<15:0>.SHOULD BE ALL 0'S.
4171 025656 013705 177754 MOV CDR,R5 ;SAVE CDR CONTENTS
4172 025662 010210 MOV R2,(R0) ;WRITE HIT CACUSES UPDATE TO CACHE DATA
4173 ;STORE LOCATION.WRITE ALL 1'S.
4174 025664 005710 TST (R0) ;READ HIT.CHICK DATA STORE BITS TO
4175 ;CDR <15:0>.SHOULD BE ALL 1'S.
4176 025666 013703 177754 MOV CDR,R3 ;SAVE CDR CONTENTS
4177 025672 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
    
```

```

4178 025674 000240      NOP      ;FOR LOOP ON ERROR
4178 025676 005705      TST      R5      ;SHOULD BE ALL 0'S
4179 025700 001424      BEQ      8$      ;PASS
4180 025702 012737 001015 177746  MOV      #OFF,CCR ;DISABLE CACHE
4181 025710 105037 177750  CLR      CMR      ;CLEAR MAINT. MODE
4182 025714 005037 047332  CLR      EXDAT6   ;SPECIFY EXPECTED CACHE DATA STORE DATA
4183 025720 010537 047330  MOV      R5,CDR150 ;SPECIFY CACHE DATA STORE DATA READ
4184                                ;THRU CDR<15:0>
4185 025724 010037 047334  MOV      R0,CA121 ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4186 025730 006237 047334  ASR      CA121
4187 025734 104406      ERROR    ;ERROR
                                ;-----
                                025736 025734      .WORD    -2
4188                                ;DATA STORE MARCH PATTERN TEST
4189                                ;READING CACHE DATA STORE DATA
4190                                ;THRU CDR<15:0> DID NOT READ ALL 0'S.
4191                                ;THIS SUGGESTS THAT A RAM LOCATION
4192                                ;SPECIFIED BY CA121 WAS OVERWRITTEN
4193                                ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
4194                                ;LOCATION.ANY BIT IN CDR150 DATA
4195                                ;THAT IS A 1 MAY POINT TO A BAD
4196                                ;CACHE DATA STORE RAM.
4197
4198 025740 047332      EXDAT6   ; EXPECTED CACHE DATA STORE DATA
4199 025742 047330      CDR150   ; CACHE DATA STORE DATA READ
4200                                ;THRU CDR<15:0>
4201 025744 047334      CA121    ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4202 025746 000000      .WORD    0
4203 025750 000435      BR       3$
4204 025752 022703 177777 8$:    CMP      #-1,R3 ;END THE TEST
4205 025756 001425      BEQ      9$      ;SHOULD BE ALL 1'S
4206 025760 012737 001015 177746  MOV      #OFF,CCR ;PASS
4207 025766 105037 177750  CLR      CMR      ;DISABLE CACHE
4208 025772 012737 177777 047332  MOV      #-1,EXDAT6 ;CLEAR MAINT. MODE
4209 026000 010337 047330  MOV      R3,CDR150 ;SPECIFY EXPECTED CACHE DATA STORE DATA
4210                                ;SPECIFY CACHE DATA STORE DATA READ
4211 026004 010037 047334  MOV      R0,CA121 ;THRU CDR<15:0>
4212 026010 006237 047334  ASR      CA121    ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4213 026014 104406      ERROR    ;ERROR
                                ;-----
                                026016 026014      .WORD    -2
4214                                ;DATA STORE MARCH PATTERN TEST
4215                                ;READING CACHE DATA STORE DATA
4216                                ;THRU CDR<15:0> DID NOT READ ALL 1'S.
4217                                ;ANY BIT IN CDR150 DATA
4218                                ;THAT IS A 0 MAY POINT TO A BAD
4219                                ;CACHE DATA STORE RAM.
4220
4221 026020 047332      EXDAT6   ; EXPECTED CACHE DATA STORE DATA
4222 026022 047330      CDR150   ; CACHE DATA STORE DATA READ
4223                                ;THRU CDR<15:0>
4224 026024 047334      CA121    ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4225 026026 000000      .WORD    0
4226 026030 000405      BR       3$
4227 026032 062700 000002 9$:    ADD      #2,R0   ;END TEST
4228 026036 022700 100000  CMP      #100000,R0 ;NEXT LOCATION
4229 026042 001304      BNE      7$      ;HAS ALL HI CACHE BEEN DONE?
                                ;NO,CONTINUE
    
```

4230	026044	012737	001015	177746	3\$:	MOV	#OFF,CCR	:DISABLE CACHE
4231	026052	105037	177750			CLRB	CMR	:DISABLE MAINT. MODE
4232	026056	000240			10\$:	NOP		:END OF TEST
	026060	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

CKKKABO 11-44 KK11B CACHE  
TEST # 150 - VERIFY CACHE TAG STORE RAM MEMORY IC'S  
4247

```
.SBTTL TEST # 150 - VERIFY CACHE TAG STORE RAM MEMORY IC'S
*****
*TEST 150 - VERIFY CACHE TAG STORE RAM MEMORY IC'S
*   VERIFY CACHE TAG STORE RAM MEMORY IC'S BY PERFORMING A
*   MARCH PATTERN TEST TO LOW CACHE AREA OF TAG STORE(LOC. 0000-3777)
*   PROCEDURE: 1. WRITE ALL 0'S TO ALL LO CACHE TAG STORE
*               RAMS CORRESPONDING TO LOCATIONS 0000-3777
*               2. READ 0'S FROM ALL LO CACHE RAMS CORRESPONDING
*               TO LOCATION 0000
*               3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
*               0000.
*               4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
*               0000.
*               5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*               AND UNTIL LOC. 3777 IS REACHED.
*****
```

4248 026064 000004  
026066 026076  
026070 070000  
026072 000000  
026074 070070  
026076 012737 001015 177746  
026104 004437 002370  
026110 026454

```
TST150:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
      .WORD 40$            ;TEST START LOCATION
      .WORD 1$-40$+67764  ;LOOP ON ERROR START LOCATION
      .WORD 0              ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR        ;DISABLE CACHE
      JSR R4,RELCTH      ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2        ;ADDRESS OF START OF NEXT TEST
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

4249 026112 005037 177752  
4250 026116 012700 060000  
4251 026122 012701 040000  
4252 026126 112737 000003 177750  
4253  
4254  
4255  
4256 026134 012737 000015 177746  
4257 026142 005721  
4258 026144 005720  
4259 026146 020027 070000  
4260 026152 001373  
4261 026154 012737 177777 177752  
4262 026162 012700 060000  
4263 026166 005710  
4264  
4265  
4266  
4267  
4268 026170 013705 177752  
4269 026174 005710  
4270  
4271 026176 013703 177752  
4272 026202 000240  
026204 000240  
4273 026206 042705 000177  
4274 026212 005705

```
1$: CLR CHR                ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
     MOV #60000,R0         ;ADDR. LOC. 60000 TO R0
     MOV #40000,R1         ;ADDR. LOC. 40000 TO R1
     MOVB #HODO+TDAR,CMR  ;HODO ALLOWS CACHE UPDATES & TAG STORE BITS
                          ;TO BE WRITTEN TO CHR<15:7> ONLY DURING
                          ;THE DESTINATION MEMORY ACCESS OF AN INSTRUCTION
                          ;TDAR ALLOWS TAG FIELD TO BE WRITTEN INTO FROM AMR<8:0>
6$: MOV #15,CCR           ;WRITE ENABLE CACHE TAG STORES
     TST (R1)+             ;WRITE ALL LOW CACHE TAG STORE WITH 0'S
     TST (R0)+
     CMP R0,#70000        ;
     BNE 6$                ;DONE?
     MOV #-1,CHR          ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
     MOV #60000,R0        ;ADDR. 60000 TO R0
     TST (R0)             ;READ MISS TO CACHE TAG STORE LOCATION
                          ;SPECIFIED BY R0.CLOCK TAG STORE
                          ;BITS INTO CHR<15:7>.SHOULD BE ALL 0'S.
                          ;ALSO CAUSES TAG STORE LOCATION TO BE
                          ;WRITTEN WITH 1'S FROM AMR<8:0>
7$: MOV CHR,R5            ;SAVE CHR CONTENTS
     TST (R0)             ;READ MISS.CLOCK TAG STORE BITS TO
                          ;CHR <15:0>.SHOULD BE ALL 1'S.
25$: MOV CHR,R3           ;SAVE CHR CONTENTS
     NOP                   ;INSTRUCTION 'JMP 1$' PLACED HERE
     NOP                   ;FOR LOOP ON ERROR
     BIC #177,R5          ;
     TST R5                ;SHOULD BE ALL 0'S
```



```

4275 026214 001437 BEQ 8$ ;PASS
4276 026216 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4277 026224 105037 177750 CLR CLRB ;CLEAR MAINT. MODE
4278 026230 005037 047350 CLR EXDAT3 ;SPECIFY EXPECTED CACHE TAG STORE DATA
4279 026234 010537 047322 MOV R5,CHR157 ;SPECIFY CACHE TAG STORE TAG READ
4280 ;THRU CHR<15:7>
4281 026240 012737 000007 002062 MOV #7,LOOP ;PREPARE CHR157 FOR TYPEOUT
4282 026246 006237 047322 4$: ASR CHR157
4283 026252 042737 100000 047322 BIC #100000,CHR157
4284 026260 005337 002062 DEC LOOP
4285 026264 001370 BNE 4$
4286 026266 010037 047334 MOV R0,CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4287 026272 006237 047334 ASR CA121
4288 026276 104406 ERROR ;ERROR
;-----
026300 026276 .WORD -2
4289 ;TAG STORE MARCH PATTERN TEST
4290 ;READING CACHE TAG STORE DATA
4291 ;THRU CHR<15:7> DID NOT READ ALL 0'S.
4292 ;THIS SUGGESTS THAT A RAM LOCATION
4293 ;SPECIFIED BY CA121 WAS OVERWRITTEN
4294 ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
4295 ;LOCATION.ANY BIT IN CHR157 DATA
4296 ;THAT IS A 1 MAY POINT TO A BAD
4297 ;CACHE TAG STORE RAM.
4298
4299 026302 047350 EXDAT3 ; EXPECTED CACHE TAG STORE DATA
4300 026304 047322 CHR157 ; CACHE TAG STORE DATA READ
4301 ;THRU CHR<15:7>
4302 026306 047334 CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4303 026310 000000 .WORD 0
4304 026312 000452 BR 3$ ;END THE TEST
4305 026314 042703 000177 8$: BIC #177,R3 ;PREPARE R3 FOR CHECK
4306 026320 022703 177600 CMP #177600,R3 ;SHOULD BE ALL 1'S
4307 026324 001440 BEQ 9$ ;PASS
4308 026326 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4309 026334 105037 177750 CLR CLRB ;CLEAR MAINT. MODE
4310 026340 012737 177777 047350 MOV #-1,EXDAT3 ;SPECIFY EXPECTED CACHE TAG STORE DATA
4311 026346 010337 047322 MOV R3,CHR157 ;SPECIFY CACHE TAG STORE DATA READ THRU CHR<15:7>
4312 026352 012737 000007 002062 MOV #7,LOOP ;PREPARE CHR157 FOR TYPEOUT
4313 026360 006237 047322 5$: ASR CHR157
4314 026364 042737 100000 047322 BIC #100000,CHR157
4315 026372 005337 002062 DEC LOOP
4316 026376 001370 BNE 5$
4317 026400 010037 047334 MOV R0,CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4318 026404 006237 047334 ASR CA121
4319 026410 104406 ERROR ;ERROR
;-----
026412 026410 .WORD -2
4320 ;TAG STORE MARCH PATTERN TEST
4321 ;READING CACHE TAG STORE DATA
4322 ;THRU CHR<15:7> DID NOT READ ALL 15'S.
4323 ;ANY BIT IN CHR157 DATA
4324 ;THAT IS A 0 MAY POINT TO A BAD
4325 ;CACHE TAG STORE RAM.
4326
4327 026414 047350 EXDAT3 ; EXPECTED CACHE TAG STORE DATA
    
```

4328									
4329									
4330	026416	047322							
4331	026420	047334							
4332	026422	000000							
4333	026424	000405							
4334	026426	062700	000002	9\$:					
4335	026432	022700	070000						
4336	026436	001253							
4337	026440	012737	001015	177746	3\$:				
4338	026446	105037	177750						
4339	026452	000240			10\$:				
	026454	005237	001472						

```

; CACHE TAG STORE DATA READ
; THRU CHR<15:7>
; CACHE TAG STORE READ THRU CHR<15:7>
; SPECIFY FAILED TAG STORE ADDRESS LOCATION

; END TEST
; NEXT LOCATION
; HAS ALL LO CACHE BEEN DONE?
; NO, CONTINUE
; DISABLE CACHE
; DISABLE MAINT. MODE
; END OF TEST
; INCREMENT TEST COUNTER

```

```

CHR157
CA121
.WORD 0
BR 3$
ADD #2,RO
CMP #70000,RO
BNE 7$
MOV #OFF,CCR
CLRB CMR
NOP
INC $TESTN

```

TEST # 151 - VERIFY CACHE TAG STORE RAM MEMORY IC'S  
4354

```

.SBTTL TEST # 151 - VERIFY CACHE TAG STORE RAM MEMORY IC'S
*****
*TEST 151 - VERIFY CACHE TAG STORE RAM MEMORY IC'S
*VERIFY CACHE TAG STORE RAM MEMORY IC'S BY PERFORMING A
*MARCH PATTERN TEST TO HIGH CACHE AREA OF TAG STORE(LOC. 4000-7777)
*PROCEDURE: 1. WRITE ALL 0'S TO ALL HI CACHE TAG STORE
*RAMS CORRESPONDING TO LOCATIONS 4000-7777
*2. READ 0'S FROM ALL HI CACHE RAMS CORRESPONDING
*TO LOCATION 4000
*3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
*4000.
*4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
*4000.
*5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*AND UNTIL LOC. 7777 IS REACHED.
*****

```

```

4355 026460 000004
      026462 026472
      026464 060000
      026466 000000
      026470 060070
      026472 012737 001015 177746
      026500 004437 002342
      026504 027050

```

```

TST151:
      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                      ;ERROR/LOOP ON TEST
                      ;TEST START LOCATION
      .WORD 40$      ;LOOP ON ERROR START LOCATION
      .WORD 1$-40$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0        ;LOOP ON ERROR END LOCATION
      .WORD 25$-40$+57764 ;DISABLE CACHE
      MOV #OFF,CCR   ;LOCATE TEST CODE TO LOW CACHE SPACE
      JSR R4,RELCTL ;ADDRESS OF START OF NEXT TEST
      .WORD 10$+2

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE

```

```

4356 026506 005037 177752
4357 026512 012700 070000
4358 026516 012701 050000
4359 026522 112737 000003 177750
4360
4361
4362
4363 026530 012737 000015 177746
4364 026536 005721
4365 026540 005720
4366 026542 020027 100000
4367 026546 001373
4368 026550 012737 177777 177752
4369 026556 012700 070000
4370 026562 005710
4371
4372
4373
4374
4375 026564 013705 177752
4376 026570 005710
4377
4378 026572 013703 177752
4379 026576 000240
      026600 000240
4380 026602 042705 000177
4381 026606 005705

```

```

1$: CLR CHR          ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
     MOV #70000,R0   ;ADDR. LOC. 70000 TO R0
     MOV #50000,R1   ;ADDR. LOC. 50000 TO R1
     MOVB #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & TAG STORE BITS
                          ;TO BE WRITTEN TO CHR<15:7> ONLY DURING
                          ;THE DESTINATION MEMORY ACCESS OF AN INSTRUCTION
                          ;TDAR ALLOWS TAG FIELD TO BE WRITTEN INTO FROM AMR<8:0>
6$: MOV #15,CCR      ;WRITE ENABLE CACHE TAG STORES
     TST (R1)+       ;WRITE ALL HIGH CACHE TAG STORE WITH 0'S
     TST (R0)+
     CMP R0,#100000  ;DONE?
     BNE 6$          ;NO
     MOV #-1,CHR     ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
     MOV #70000,R0   ;ADDR. 70000 TO R0
     TST (R0)        ;READ MISS TO CACHE TAG STORE LOCATION
                          ;SPECIFIED BY R0.CLOCK TAG STORE
                          ;BITS INTO CHR<15:7>.SHOULD BE ALL 0'S.
                          ;ALSO CAUSES TAG STORE LOCATION TO BE
                          ;WRITTEN WITH 1'S FROM AMR<8:0>
7$: MOV CHR,R5       ;SAVE CHR CONTENTS
     TST (R0)        ;READ MISS.CLOCK TAG STORE BITS TO
                          ;CHR <15:7>.SHOULD BE ALL 1'S.
25$: MOV CHR,R3      ;SAVE CHR CONTENTS
     NOP             ;INSTRUCTION 'JMP 1$' PLACED HERE
     NOP             ;FOR LOOP ON ERROR
     BIC #177,R5     ;SHOULD BE ALL 0'S
     TST R5

```

```

TEST # 151 - VERIFY CACHE TAG STORE RAM MEMORY IC'S
4382 026610 001437 BEQ 8$ ;PASS
4383 026612 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4384 026620 105037 177750 CLRB CMR ;CLEAR MAINT. MODE
4385 026624 005037 047350 CLR EXDAT3 ;SPECIFY EXPECTED CACHE TAG STORE DATA
4386 026630 010537 047322 MOV R5,CHR157 ;SPECIFY CACHE TAG STORE TAG READ
4387 ;THRU CHR<15:7>
4388 026634 012737 000007 002062 MOV #7,LOOP ;PREPARE CHR157 FOR TYPEOUT
4389 026642 006237 047322 4$: ASR CHR157
4390 026646 042737 100000 047322 BIC #100000,CHR157
4391 026654 005337 002062 DEC LOOP
4392 026660 001370 BNE 4$
4393 026662 010037 047334 MOV R0,CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4394 026666 006237 047334 ASR CA121
4395 026672 104406 ERROR ;ERROR
;-----
026674 026672 .WORD .-2
4396 ;TAG STORE MARCH PATTERN TEST
4397 ;READING CACHE TAG STORE DATA
4398 ;THRU CHR<15:7> DID NOT READ ALL 0'S.
4399 ;THIS SUGGESTS THAT A RAM LOCATION
4400 ;SPECIFIED BY CA121 WAS OVERWRITTEN
4401 ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
4402 ;LOCATION.ANY BIT IN CHR157 DATA
4403 ;THAT IS A 1 MAY POINT TO A BAD
4404 ;CACHE TAG STORE RAM.
4405
4406 026676 047350 EXDAT3 ; EXPECTED CACHE TAG STORE DATA
4407 026700 047322 CHR157 ; CACHE TAG STORE DATA READ
4408 ;THRU CHR<15:7>
4409 026702 047334 CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4410 026704 000000 .WORD 0
4411 026706 000452 BR 3$ ;END THE TEST
4412 026710 042703 000177 8$: BIC #177,R3 ;PREPARE R3 FOR CHECK
4413 026714 022703 177600 CMP #177600,R3 ;SHOULD BE ALL 1'S
4414 026720 001440 BEQ 9$ ;PASS
4415 026722 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4416 026730 105037 177750 CLRB CMR ;CLEAR MAINT. MODE
4417 026734 012737 177777 047350 MOV #-1,EXDAT3 ;SPECIFY EXPECTED CACHE TAG STORE DATA
4418 026742 010337 047322 MOV R3,CHR157 ;SPECIFY CACHE TAG STORE DATA READ THRU CHR<15:7>
4419 026746 012737 000007 002062 MOV #7,LOOP ;PREPARE CHR157 FOR TYPEOUT
4420 026754 006237 047322 5$: ASR CHR157
4421 026760 042737 100000 047322 BIC #100000,CHR157
4422 026766 005337 002062 DEC LOOP
4423 026772 001370 BNE 5$
4424 026774 010037 047334 MOV R0,CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4425 027000 006237 047334 ASR CA121
4426 027004 104406 ERROR ;ERROR
;-----
027006 027004 .WORD .-2
4427 ;TAG STORE MARCH PATTERN TEST
4428 ;READING CACHE TAG STORE DATA
4429 ;THRU CHR<15:7> DID NOT READ ALL 15'S.
4430 ;ANY BIT IN CHR157 DATA
4431 ;THAT IS A 0 MAY POINT TO A BAD
4432 ;CACHE TAG STORE RAM.
4433
4434 027010 047350 EXDAT3 ; EXPECTED CACHE TAG STORE DATA

```

4435																			: CACHE TAG STORE DATA READ
4436																			: THRU CHR<15:7>
4437	027012	047322																	: CACHE TAG STORE READ THRU CHR<15:7>
4438	027014	047334																	: SPECIFY FAILED TAG STORE ADDRESS LOCATION
4439	027016	000000																	
4440	027020	000405																	: END TEST
4441	027022	062700	000002		9%:														: NEXT LOCATION
4442	027026	022700	100000																: HAS ALL HI CACHE BEEN DONE?
4443	027032	001253																	: NO, CONTINUE
4444	027034	012737	001015	177746	3%:														: DISABLE CACHE
4445	027042	105037	177750																: DISABLE MAINT. MODE
4446	027046	000240			10%:														: END OF TEST
	027050	005237	001472																: INCREMENT TEST COUNTER

```

.SBTTL TEST # 152 - VERIFY THAT BYTE DATA PARITY STORES CAN HOLD A 0
*****
*TEST 152 - VERIFY THAT BYTE DATA PARITY STORES CAN HOLD A 0
*VERIFY THAT LOW AND HI BYTE DATA PARITY STORES CAN HOLD A 0 AT DATA
*PARITY STORE LOCATION 0000.
*PROCEDURE: GENERATE 0'S FROM UPPER AND LOWER BYTE PARITY
*DATA GENERATORS BY PLACING ALL 0'S ON INPUTS.
*ZERO'S WILL THEN BE WRITTEN INTO DATA PARITY STORE
*LOCATION 0000.READ DATA PARITY STORE BITS FROM
*CMR<11:10>
*CONDITIONS:INPUTS TO DATA PARITY GEN:
*WRTD<15:0> ALL 0'S
*WVPD(1)= 0
*DATA PARITY STORE ADDRESS:
*CA<12:1>=0000
*RESULT: CMR<11:10> BOTH 0
*****

```

```

4463 027054 000004          TST152:          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
027056 027066          .WORD 40$          ;LOOP ON ERROR START LOCATION
027060 070004          .WORD 1$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
027062 000000          .WORD 0            ;LOOP ON ERROR END LOCATION
027064 070040          .WORD 25$-40$+67764 ;DISABLE CACHE
027066 012737 001015 177746 40$: MOV #OFF,CCR        ;LOCATE TEST CODE TO HIGH CACHE SPACE
027074 004437 002370          JSR R4,RELCTH      ;ADDRESS OF START OF NEXT TEST
027100 027212          .WORD 10$+2

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

4464 027102 005037 060000          CLR 60000          ;0'S TO MAIN MEMORY LOCATION
4465 027106 112737 000002 177750 1$: MOVB #HODO,CMR    ;ALLOWS UPPER AND LOWER BYTE DATA
4466          ;PARITY STORE BITS TO BE WRITTEN TO
4467          ;CMR<11:10> ONLY DURING THE DESTINATION
4468          ;ACCESS OF AN INSTRUCTION.
4469 027114 012737 000015 177746          MOV #15,CCR        ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
4470 027122 005737 040000          TST 40000          ;
4471 027126 005737 060000          TST 60000          ;PLACE ALL 0'S ON WRTD<15:0> INPUTS
4472          ;THEREBY WRITING 0 INTO PARITY STORE LOCATION 0000.
4473 027132 005737 060000          TST 60000          ;WRITE UPPER AND LOWER DATA PARITY BITS FROM
4474          ;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
4475 027136 013701 177750          MOV CMR,R1         ;SAVE CMR DATA
4476 027142 00C240          NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
4477 027144 000240          NOP                ;FOR LOOP ON ERROR
4477 027146 105037 177750          CLRB CMR           ;DISABLE MAINT. MODE
4478 027152 012737 001015 177746          MOV #OFF,CCR      ;DISABLE CACHE
4479 027160 032701 004000          BIT #HPB,R1       ;CHECK FOR 0
4480 027164 001403          BEQ 9$             ;PASS
4481 027166 104406          ERROR            ;ERROR
                                ;-----
                                ;DATA PARITY GEN. & STORE TESTS
                                ;READING CACHE MAINT. REGISTER
                                ;BIT 11 FOR UPPER BYTE PARITY DATA DID
                                ;NOT RESULT IN 0.
027170 027166          .WORD -2
4482          ;
4483          ;
4484          ;
4485          ;
4486 027172 000000          .WORD 0

```

4487	027174	032701	002000	9\$:	BIT	#LPB,R1	:CHECK 0 FOR LOWER BYTE PARITY DATA
4488	027200	001403			BEQ	10\$	:PASS
4489	027202	104406			ERROR		:ERROR
	027204	027202			.WORD	.-2	:-----
4490							:DAT. PARITY GEN. \$ STORE TESTS
4491							:READING CACHE MAINT. REGISTER
4492							:BIT 10 FOR LOWER BYTE PARITY DATA DID
4493							:NOT RESULT IN 0.
4494	027206	000000			.WORD	0	
4495	027210	000240		10\$:	NOP		:END OF TEST
	027212	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

TEST # 153 - CHK THAT LOW BYTE DATA PARITY GEN WRITES A 1  
4511

```

.SBTTL TEST # 153 - CHK THAT LOW BYTE DATA PARITY GEN WRITES A 1
*****
*TEST 153 - CHK THAT LOW BYTE DATA PARITY GEN WRITES A 1
*   VERIFY THAT THE LOW BYTE DATA PARITY GENERATOR WILL WRITE A 1 INTO ADDRESS LOCATION
*   0000 FOR FLOATING 1 ACROSS 0 DATA PATTERN ON DATA PARITY GENERATOR INPUTS.
*   PROCEDURE: FOR EACH FLOATING 1 PATTERN READ DATA PARITY STORE BITS
*               FROM CMR<11:10>
*
*   CONDITIONS:
*       INPUTS TO DATA PARITY GEN.:
*           WRTD<7:0> FLOATING 1 ACROSS 0'S
*           WRTD<15:8> ALL 0'S
*           WWPD(1)=0
*       DATA PARITY STORE ADDRESS:
*           CA<12:1>=0000
*
*   RESULT: CMR<11>=0
*           CMR<10>=1
*****

```

```

4512 027216 000004
      027220 027230
      027222 070014
      027224 000000
      027226 070050
      027230 012737 001015 177746
      027236 004437 002370
      027242 027404

```

```

TST153:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
      .WORD 40$             ;TEST START LOCATION
      .WORD 1$-40$+67764   ;LOOP ON ERROR START LOCATION
      .WORD 0               ;SCOPE SYNC. NOT IMPL. MENTED FOR THIS TEST
      .WORD 25$-40$+67764  ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR         ;DISABLE CACHE
      JSR R4,RELCTH        ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2          ;ADDRESS OF START OF NEXT TEST

```

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

```

4513 027244 012737 000001 047344
4514 027252 013737 047344 060000 2$:
4515 027260 112737 000002 177750 1$:
4516
4517
4518
4519 027266 012737 000015 177746
4520
4521 027274 005737 040000
4522 027300 005737 060000
4523
4524
4525 027304 005737 060000
4526
4527 027310 013701 177750
4528 027314 000240
      027316 000240
4529 027320 105037 177750
4530 027324 012737 001015 177746
4531 027332 032701 004000
4532 027336 001404
4533 027340 104406

```

```

      MOV #1,FLTPAT        ;1ST FLOATING 1 PATTERN:000001
      MOV FLTPAT,60000     ;FLOATING PATTERN TO MAIN MEMORY
      MOVB #HODO,CMR       ;ALLOWS UPPER AND LOWER BYTE DATA
                          ;PARITY STORE BITS TO BE WRITTEN TO
                          ;CMR<11:10> ONLY DURING THE DESTINATION
                          ;ACCESS OF AN INSTRUCTION.
      MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE PARITY
                          ;STORE
      TST 40000
      TST 60000            ;PLACE FLOATING 1 PATTERN ON WRTD<15:0> INPUTS
                          ;THEREBY WRITING 1 IN LOW BYTE AND 0 IN HI
                          ;BYTE DATA PARITY STORE LOCATION 0000.
      TST 60000            ;WRITE UPPER AND LOWER BYTE DATA PARITY BITS FROM
                          ;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
      MOV CMR,R1           ;SAVE CMR DATA
      NOP                  ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP                  ;FOR LOOP ON ERROR
      CLRB CMR             ;DISABLE MAINT. MODE
      MOV #OFF,CCR        ;DISABLE CACHE
      BIT #HPB,R1         ;CHECK 0 FOR UPPER BYTE PARITY STORE
      BEQ 8$              ;PASS
      ERROR                ;ERROR
      -----

```

```

      027342 027340
      .WORD -.2
4534
4535

```

:DATA PARITY GEN. & STORE TESTS  
:READING CACHE MAINT. REGISTER



```

4536                                     ;BIT 11 FOR UPPER BYTE PARITY DATA DID
4537                                     ;NOT RESULT IN 0.
4538 027344 047344                     FLTPAT      ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
4539                                     ;GENERATOR INPUTS: WRTD<15:0>
4540 027346 000000                       .WORD      0
4541 027350 032701 002000                8$: BIT    #LPB,R1      ;CHECK 1 FOR LOWER BYTE PARITY DATA
4542 027354 001004                       BNE      9$          ;PASS
4543 027356 104406                       ERROR     ;ERROR
                                     ;-----
                                027360 027356     .WORD     .-2
4544                                     ;DAT. PARITY GEN. & STORE TESTS
4545                                     ;READING CACHE MAINT. REGISTER
4546                                     ;BIT 10 FOR LOWER BYTE PARITY DATA DID
4547                                     ;NOT RESULT IN 1.
4548 027362 047344                     FLTPAT      ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
4549                                     ;GEN. INPUT: WRTD<15:0>
4550 027364 000000                       .WORD      0
4551 027366 006337 047344                9$: ASL    FLTPAT     ;NEXT PATTREN
4552 027372 032737 000400 047344        BIT    #400,FLTPAT ;HAS PATTERN 000200 BEEN DONE
4553 027400 001724                       BEQ      2$          ;NO
4554 027402 000240                       10$: NOP           ;END OF TEST
                                027404 005237 001472 INC    $TESTN      ;INCREMENT TEST COUNTER

```

4570

```

.SBTTL TEST # 154 - CHK THAT HI BYTE DATA PARITY GEN WRITES A 1
*****
*TEST 154 - CHK THAT HI BYTE DATA PARITY GEN WRITES A 1
*VERIFY THAT THE HI BYTE DATA PARITY GENERATOR WILL WRITE A 1 INTO ADDRESS LOCATION
*FOR FLOATING 1 ACROSS 0 DATA PATTERN ON DATA PARITY GENERATOR INPUTS.
*PROCEDURE: FOR EACH FLOATING 1 PATTERN READ DATA PARITY STORE BITS
*FROM CMR<11:10>
*CONDITIONS:
*INPUTS TO DATA PARITY GEN.:
*WRTD<7:0> ALL 0'S
*WRTD<15:8> FLOATING 1 PATTERN
*WVPD(1)=0
*DATA PARITY STORE ADDRESS:
*CA<12:1>=0000
*RESULT: CMR<11>=1
*CMR<10>=0
*****

```

```

027410
4571 027410 000004
027412 027422
027414 070014
027416 000000
027420 070050
027422 012737 001015 177746
027430 004437 002370
027434 027570

```

```

TST154:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;DISABLE CACHE
;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

```

```

4572 027436 012737 000400 047344
4573 027444 013737 047344 060000
4574 027452 112737 000002 177750
4575
4576
4577
4578 027460 012737 000015 177746
4579 027466 005737 040000
4580 027472 005737 060000
4581
4582

```

```

MOV #400,FLTPAI ;1ST FLOATING 1 PATTERN:000400
MOV FLTPAT,60000 ;FLOATING PATTERN TO MAIN MEMORY
MOVWB #HODO,CMR ;ALLOWS UPPER AND LOWER BYTE DATA
;PARITY STORE BITS TO BE WRITTEN TO
;CMR<11:10> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;NO UCB SO AS TO WRITE ENABLE PARITY STORE
;
;PLACE FLOATING 1 PATTERN ON WRTD<15:0> INPUTS
;THEREBY WRITING 0 IN LOW BYTE AND 1 IN HI
;BYTE DATA PARITY STORE LOCATION 0000.
;WRITE UPPER AND LOWER BYTE DATA PARITY BITS FROM

```

000058BT027472R01100073RES060000

```

4584 027502 013701 177750
4585 027506 000240
027510 000240
4586 027512 105037 177750
4587 027516 012737 001015 177746
4588 027524 032701 004000
4589 027530 001004
4590 027532 104406
027534 027532

```

```

MOV CMR,R1 ;SAVE CMR DATA
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
BIT #HPB,R1 ;CHECK 1 FOR UPPER BYTE PARITY STORE
BNE 8$ ;PASS
ERROR ;ERROR
;-----
.WORD .-2

```



TEST # 155 - VERIFY WRITE WRONG PARITY TO BYTES DATA PARITY  
4622

```

.SBTTL TEST # 155 - VERIFY WRITE WRONG PARITY TO BYTES DATA PARITY
*****
*TEST 155 - VERIFY WRITE WRONG PARITY TO BYTES DATA PARITY
*VERIFY WRITE WRONG PARITY TO UPPER AND LOWER BYTE DATA PARITY
*STORE
*CONDITIONS:
*   INPUTS TO DATA PARITY GEN:
*   WRTD<15:0> ALL 0'S
*   WWP(1)= 1
*   DATA PARITY STORE ADDRESS:
*   CA<12:1>=0000
*RESULT: CMR<11:10> BOTH 1
*****

```

```

4623 027574 000004
      027576 027606
      027600 070004
      027602 000000
      027604 070040
      027606 012737 001015 177746
      027614 004437 002370
      027620 027750

```

```

TST155:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        ;TEST START LOCATION
        ;LOOP ON ERROR START LOCATION
        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        ;LOOP ON ERROR END LOCATION
        ;DISABLE CACHE
        ;LOCATE TEST CODE TO HIGH CACHE SPACE
        ;ADDRESS OF START OF NEXT TEST

```

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

```

4624 027622 005037 060000
4625 027626 112737 000002 177750 1$:
4626
4627
4628
4629 027634 012737 000115 177746
4630
4631 027642 005737 040000
4632 027646 005737 060000
4633
4634
4635 027652 005737 060000
4636
4637 027656 013701 177750
4638 027662 000240
      027664 000240
4639 027666 012737 001015 177746
4640 027674 105037 177750
4641 027700 052737 000400 177746
4642
4643 027706 032737 010000 177746 200$:
4644 027714 001374
4645 027716 032701 004000
4646 027722 001003
4647 027724 104406

```

```

CLR 60000 ;0'S TO MAIN MEMORY LOCATION
MOV #HODO,CMR ;ALLOWS UPPER AND LOWER BYTE DATA
;PARITY STORE BITS TO BE WRITTEN TO
;CMR<11:10> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
MOV #15+WWPD,CLR ;NO UCB SO AS TO WRITE ENABLE PARITY.
;ENABLE WRITE WRONG PARITY DATA
;
TST 40000 ;PLACE ALL 0'S ON WRTD<15:0> INPUTS
TST 60000 ;SINCE WWP IS INVOKED A 1 WILL BE
;WRITTEN INTO PARITY STORE LOCATION 0000.
;WRITE UPPER AND LOWER DATA PARITY BITS FROM
;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
;SAVE CMR DATA
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
MOV #OFF,CCR ;DISABLE CACHE
CLRB CMR ;DISABLE MAINT. MODE
BIS #FC,CCR ;BEFORE LEAVING TEST FLUSH CACHE TO
;REMOVE ANY EFFECTS OF WWP
;WAIT TILL DONE
MOV #VCIP,CCR
BNE 20C$
BIT #HPB,R1 ;CHECK 1 FOR UPPER BYTE PARITY STORE.
BNE 9$ ;PASS
ERROR ;ERROR
;-----

```

```

      027726 027724
      .WORD -.2
4648
4649
4650

```

```

;DATA PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 11 FOR UPPER BYTE PARITY DATA DID

```



```

.SBTTL TEST # 156 - VERIFY THAT TAG PARITY STORE CAN HOLD A 0
*****
*TEST 156 - VERIFY THAT TAG PARITY STORE CAN HOLD A 0
*VERIFY THAT TAG PARITY STORE CAN HOLD A 0 AT TAG
*PARITY STORE LOCATION 0000.
*PROCEDURE: GENERATE 0 FROM TAG PARITY
*GENERATOR BY PLACING ALL 0'S ON INPUTS.
*ZERO WILL BE WRITTEN INTO TAG PARITY STORE
*LOCATION 0000.READ TAG PARITY STORE BIT FROM
*CMR<9>
*CONDITIONS:INPUTS TO TAG PARITY GEN:
*TAG WRTD<21:13> ALL 0'S
*WPT(1)= 0
*TAG PARITY STORE ADDRESS:
*CA<12:1>=0000
*RESULT: CMR<9>= 0
*****
  
```

```

027754
4678 027754 000004
027756 027766
027760 070004
027762 000000
027764 070040
027766 012737 001015 177746
027774 004437 002370
030000 030076
  
```

```

TST156:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;DISABLE CACHE
;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
  
```

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
 :ARE RELOCATED TO HI CACHE SPACE

```

4679 030002 005037 177752
4680 030006 112737 000003 177750
4681
4682
4683
4684
4685
4686 030014 012737 000015 177746
4687
4688 030022 005737 040000
4689 030026 005737 060000
4690
4691 030032 005737 060000
4692 030036 013701 177750
4693 030042 000240
030044 000240
4694 030046 105037 177750
4695 030052 012737 001015 177746
4696 030060 032701 001000
4697 030064 001403
4698 030066 104406
  
```

```

1$: CLR CHR ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
MOV #HODO+TDAR,CMR ; HODO ALLOWS TAG
;PARITY STORE BIT TO BE WRITTEN TO
;CMR<9> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;TDAR ALLOWS INPUTS TO TAG PARITY STORE
;GENERATOR TO BE LOADED FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
25$: TST 40000
TST 60000 ;PLACE ALL 0'S ON TAG WRTD<21:13> INPUTS
;THEREBY WRITING 0 INTO PARITY STORE LOCATION 0000.
;WRITE TAG PARITY BITS FROM LOCAT. 0000 INTO CMR<9>
;SAVE CMR DATA
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
CLR CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
BIT #TPB,R1 ;CHECK FOR 0
BEQ 10$ ;PASS
ERROR ;ERROR
-----
  
```

```

030070 030066
4699
4700
4701
  
```

```

;TAG PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 9 FOR TAG PARITY DATA DID
  
```

4702  
4703 030072 000000  
4704 030074 000240  
          030076 005237 001472

10\$:      .WORD 0  
          NOP  
          INC     \$TESTN

;NOT RESULT IN 0.  
;END OF TEST  
;INCREMENT TEST COUNTER

```

.SBTTL TEST # 157 - VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1
*****
*TEST 157 - VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1
*VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1 INTO TAG PARITY STORE
*ADDRESS 0000 FOR FLOATING 1 PATTERN ON TAG PARITY GENERATOR INPUTS
*PROCEDURE: GENERATE 1 FROM TAG PARITY
*GENERATOR BY PLACING FLOATING 1 PATTERN ON INPUTS
*AND WRITING 1 INTO TAG PARITY STORE
*LOCATION 0000.READ TAG PARITY STORE BIT FROM
*CMR<9>
*
*CONDITIONS:
*INPUTS TO TAG PARITY GEN:
*TAG WRTD<21:13> FLOATING 1 PATTERN
*WWPD(1)= 0
*TAG PARITY STORE ADDRESS:
*CA<12:1>=0000
*
*RESULT:
*CMR<9>= 1
*****
  
```

```

4722 030102 000004
      030104 030114
      030106 070014
      030110 000000
      030112 070050
      030114 012737 001015 177746
      030122 004437 002370
      030126 030252
  
```

```

TST157:
      SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      ;ERROR/LOOP ON TEST
      .WORD 40$ ;TEST START LOCATION
      .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
      .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR ;DISABLE CACHE
      JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
  
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

```

4723 030130 012737 000001 047344
4724 030136 013737 047344 177752 2$:
4725
4726 030144 112737 000003 177750 1$:
4727
4728
4729
4730
4731
4732 030152 012737 000015 177746
4733
4734 030160 005737 040000
4735 030164 005737 060000
4736
4737 030170 005737 060000
4738 030174 013701 177750
4739 030200 000240
      030202 000240
4740 030204 105037 177750
4741 030210 012737 001015 177746
4742 030216 032701 001000
4743 030222 001004
4744 030224 104406
      030226 030224
  
```

```

      MOV #1,FLTPAT ;1ST FLOATING PATTERN
      MOV FLTPAT,CHR ;LOAD AMR<8:0> BY WRITING FLOATING
      ;PATTERN TO CHR<8:0>.
      MOV #HODO+TDAR,CMR ; HODO ALLOWS TAG
      ;PARITY STORE BIT TO BE WRITTEN TO
      ;CMR<9> ONLY DURING THE DESTINATION
      ;ACCESS OF AN INSTRUCTION.
      ;TDAR ALLOWS INPUTS TO TAG PARITY STORE
      ;GENERATOR TO BE LOADED FROM AMR<8:0>
      ;NO UCB SO AS TO WRITE ENABLE PARITY
      ;STORE
      ;
      ;PLACE FLOATING 1 PATTERN ON TAG WRTD<21:13> INPUTS
      ;THEREBY WRITING 1 INTO PARITY STORE LOCATION 0000.
      ;WRITE TAG PARITY BIT FROM LOCAT. 0000 INTO CMR<9>
      MOV CMR,R1 ;SAVE CMR DATA
      NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP ;FOR LOOP ON ERROR
      CLRB CMR ;DISABLE MAINT. MODE
      MOV #OFF,CCR ;DISABLE CACHE
      BIT #TPB,R1 ;CHECK FOR 1
      BNE 9$ ;PASS
      ERROR ;ERROR
      ;-----
      .WORD -2
  
```



4745  
4746  
4747  
4748  
4749 030230 047344      FLTPAT  
4750  
4751 030232 000000      .WORD 0  
4752 030234 006337 047344      9\$: ASL FLTPAT  
4753 030240 032737 001000 047344      BIT #1000,FLTPAT  
4754 030246 001733      BEQ 2\$  
4755 030250 000240      10\$: NOP  
         030252 005237 001472      INC \$TESTN

:TAG PARITY GEN. & STORE TESTS  
:READING CACHE MAINT. REGISTER  
:BIT 9 FOR TAG PARITY DATA DID  
:NOT RESULT IN 1.  
:PRINT FLOATING 1 PATTERN USED ON  
:TAG PARITY GEN. INPUTS: TAG WRD<21:13>  
  
:NEXT PATTERN  
:HAS PATTERN 400 BEEN DONE  
:NO,CONTINUE  
:END OF TEST  
:INCREMENT TEST COUNTER

i

4766

```

.SBTTL TEST # 160 - VERIFY WRITE WRONG PARITY TO TAG PARITY STORE
*****
*TEST 160 - VERIFY WRITE WRONG PARITY TO TAG PARITY STORE
*VERIFY WRITE WRONG PARITY TO TAG PARITY STORE
*CONDITIONS:
*   INPUTS TO TAG PARITY GEN:
*   TAG WRTO<21:13> ALL 0'S
*   WWPT(1)= 1
*   TAG PARITY STORE ADDRESS:
*   CA<12:1>=0000
*RESULT:   CMR<9>= 1
*****

```

```

4767 030256 000004
      030260 030270
      030262 070004
      030264 000000
      030266 070040
      030270 012737 001015 177746
      030276 004437 002370
      030302 030416

```

```

TST160:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        ;TEST START LOCATION
        ;LOOP ON ERROR START LOCATION
        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        ;LOOP ON ERROR END LOCATION
        ;DISABLE CACHE
        ;LOCATE TEST CODE TO HIGH CACHE SPACE
        ;ADDRESS OF START OF NEXT TEST

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

```

```

4768 030304 005037 177752
4769 030310 112737 000003 177750 1$:
4770
4771
4772
4773
4774
4775 030316 012737 002015 177746
4776
4777 030324 005737 040000
4778 030330 005737 060000
4779
4780
4781 030334 005737 060000
4782 030340 013701 177750
4783 030344 000240
      030346 000240
4784 030350 012737 001015 177746
4785 030356 105037 177750
4786 030362 052737 000400 177746
4787
4788 030370 032737 010000 177746 200$:
4789 030376 001374
4790 030400 032701 001000
4791 030404 001003
4792 030406 104406

```

```

CLR    CHR ;LOAD AMR<8:0> WITH 0'S BY WRITING TO CHR<8:0>
MOV    #HODO+TDAR,CMR ;HODO  ALLOWS TAG
        ;PARITY STORE BITS TO BE WRITTEN TO
        ;CMR<9> ONLY DURING THE DESTINATION
        ;ACCESS OF AN INSTRUCTION.
        ;TDAR ALLOWS INFJTS TO TAG PARITY GEN.
        ;TO BE LOADED FROM AMR<8:0>
        ;NO UCB SO AS TO WRITE ENABLE PARITY.
        ;ENABLE WRITE WRONG PARITY TAG
        ;
        ;PLACE ALL 0'S ON TAG WRTO<20:13> INPUTS
        ;SINCE WWPT IS INVOKED A 1 WILL BE
        ;WRITTEN INTO PARITY STORE LOCATION 0000.
        ;WRITE TAG BIT FROM LOCAT. 0000 INTO CMR<9>
        ;SAVE CMR DATA
        ;INSTRUCTION 'JMP 1$' PLACED HERE
        ;FOR LOOP ON ERROR
        ;DISABLE CACHE
        ;DISABLE MAINT. MODE
        ;BEFORE LEAVING TEST FLUSH CACHE
        ;TO REMOVE ANY EFFECTS OF WWPT
        ;WAIT TILL DONE
        ;
        ;CHECK 1 FOR TAG PARITY STORE.
        ;PASS
        ;ERROR
        ;-----

```

```

030410 030406 .WORD  -2

```

4793  
4794  
4795  
4796  
4797 030412 000000  
4798 030414 000240  
030416 005237 001472

10\$: .WORD 0  
NOP  
INC \$TESTN

:TAG PARITY GEN. & STORE TESTS  
:READING CACHE MAINT. REGISTER  
:BIT 9 FOR TAG PARITY DATA DID  
:NOT RESULT IN 1.  
:END OF TEST  
:INCREMENT TEST COUNTER

4812

```

.SBTTL TEST # 161 - CLEAR ALL LOW CACHE DATA & TAG PARITY STORES
*****
*TEST 161 - CLEAR ALL LOW CACHE DATA & TAG PARITY STORES
*WRITE AND READ 0'S TO ALL LOW CACHE DATA PARITY AND TAG PARITY STORES
*CONDITIONS:
*   INPUTS TO DATA PARITY GEN:
*       WRD<15:0> ALL 0'S
*       WUPD(1)=0
*   INPUTS TO TAG PARITY GEN.:
*       TAG WRD<21:13> ALL 0'S
*       WUPT(1)=0
*   DATA PARITY/TAG PARITY STORE ADDRESS:
*       CA<12:1>=0000 TO 3777
*RESULT:   CMR<11:9> ALL 0
*****

```

4813 030422 000004

```

030424 030434
030426 070030
030430 000000
030432 070056
030434 012737 001015 177746
030442 004437 002370
030446 030646

```

```

TST161:
      SPCOND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
                          ;TEST START LOCATION
      .WORD 40$             ;LOOP ON ERROR START LOCATION
      .WORD 1$-40$+67764   ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0               ;LOOP ON ERROR END LOCATION
      .WORD 25$-40$+67764  ;DISABLE CACHE
      MOV #OFF,CCR         ;LOCATE TEST CODE TO HIGH CACHE SPACE
      JSR R4,RELCTH       ;ADDRESS OF START OF NEXT TEST
      .WORD 10$+2

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

```

```

4814 030450 005037 177752
4815 030454 012705 060000
4816 030460 005025
4817 030462 020527 070000
4818 030466 001374
4819 030470 012705 060000
4820 030474 012703 040000
4821 030500 112737 000003 177750
4822
4823
4824
4825
4826
4827 030506 012737 000015 177746
4828 030514 005713
4829 030516 005715
4830
4831 030520 005715
4832 030522 013701 177750
4833 030526 000240
      030530 000240
4834 030532 105037 177750
4835 030536 012737 001015 177746
4836 030544 010537 047334
4837 030550 006237 047334
4838 030554 032701 004000

```

```

      CLR CHR                ;LOAD AMR<8:0> WITH ALL 0'S
      MOV #60000,R5         ;ADDRESS 60000 TO R5
      CLR (R5)+             ;CLEAR ALL LOW CACHE MAIN MEMORY
      CMP R5,#70000
      BNE 2$
      MOV #60000,R5         ;1ST ADDRESS LOCATION IN R5
      MOV #40000,R3        ;ADDRESS 40000 IN R3
      MOVB #HODO+TDAR,CMR  ;HODO ALLOWS DATA PARITY/TAG PARITY
                          ;STORE BITS TO BE WRITTEN TO
                          ;CMR<11:9> ONLY DURING THE DESTINATION
                          ;ACCESS OF AN INSTRUCTION.
                          ;TDAR ALLOWS TAG PARITY STORE GENERATOR
                          ;INPUTS TO BE LOADED FROM AMR<8:0>
                          ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
      MOV #15,CCR
      TST (R3)
      TST (R5)
      TST (R5)
      MOV CMR,R1           ;WRITE 0'S INTO DATA/TAG PARITY STORE
                          ;ADDRESS LOCATION SPECIFIED BY R5
                          ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
      NOP                  ;SAVE CMR DATA
      NOP                  ;INSTRUCTION 'JMP 1$' PLACED HERE
      CLRB CMR            ;FOR LOOP ON ERROR
                          ;DISABLE MAINT. MODE
      MOV #OFF,CCR        ;DISABLE CACHE
      MOV R5,CA121        ;GET PARITY ADDRESS LOCATION USED
      ASR CA121           ;PREPARE CA121 FOR TYPEOUT
      BIT #HPB,R1        ;CHECK 0 HI BYTE PARITY STORE

```

4839	030560	001404				BEQ	7\$		:PASS
4840	030562	104406				ERROR			:ERROR
	030564	030562				.WORD	.-2		:-----
4841									:DATA/TAG PARITY GEN. & STORE TESTS
4842									:READING CACHE MAINT. REGISTER
4843									:BIT 11 FOR UPPER BYTE PARITY DATA DID
4844									:NOT RESULT IN 0.
4845	030566	047334				CA121			:PRINT PARITY STORE ADDRESS LOCATION
4846									:USED: CA<12:1>
4847	030570	000000				.WORD	0		
4848	030572	032701	002000		7\$:	BIT	#LPB,R1		:CHECK 0 FOR LOWER BYTE PARITY DATA
4849	030576	001404				BEQ	8\$		:PASS
4850	030600	104406				ERROR			:ERROR
	030602	030600				.WORD	.-2		:-----
4851									:DATA/TAG PARITY GEN. & STORE TESTS
4852									:READING CACHE MAINT. REGISTER
4853									:BIT 10 FOR LOWER BYTE PARITY DATA DID
4854									:NOT RESULT IN 0.
4855	030604	047334				CA121			:PRINT PARITY STORE ADDRESS USED:CA<12:1>
4856	030606	000000				.WORD	0		
4857	030610	032701	001000		8\$:	BIT	#TPB,R1		:CHECK 0 FOR TAG PARITY DATA
4858	030614	001404				BEQ	9\$		:PASS
4859	030616	104406				ERROR			:ERROR
	030620	030616				.WORD	.-2		:-----
4860									:DATA/TAG PARITY GEN. AND STORAGE TESTS
4861									:READING CACHE MAINT.REGISTER BIT 9 FOR
4862									:TAG PARITY DATA DID NOT RESULT IN 0.
4863	030622	047334				CA121			:PRINT PARITY STORE ADDRESS USED: CA<12:1>
4864	030624	000000				.WORD	0		
4865	030626	062705	000002		9\$:	ADD	#2,R5		:NEXT PARITY STORE ADDRESS LOCATION
4866	030632	062703	000002			ADD	#2,R3		
4867	030636	022705	070000			CMP	#70000,R5		:HAVE ALL LOW CACHE PARITY STORE ADDRESS
4868									:LOCATIONS BEEN DONE
4869	030642	001316				BNE	1\$		:NO,CONTINUE
4870	030644	000240			10\$:	NOP			:END OF TEST
	030646	005237	001472			INC	\$TESTN		:INCREMENT TEST COUNTER

```

.SBTTL TEST # 162 - CLEAR ALL CACHE DATA & TAG PARITY STORES
*****
*TEST 162 - CLEAR ALL CACHE DATA & TAG PARITY STORES
*WRITE AND READ 0'S TO ALL HI CACHE DATA PARITY AND TAG PARITY STORES
*CONDITIONS:
*   INPUTS TO DATA PARITY GEN:
*     WRD<15:0> ALL 0'S
*     WRPD(1)= 0
*   INPUTS TO TAG PARITY GEN.:
*     TAG WRD<21:13> ALL 0'S
*     WRPD(1)=0
*   DATA PARITY/TAG PARITY STORE ADDRESS:
*     CA<12:1>=4000 TO 7777
*RESULT:   CMR<11:9> ALL 0
*****

```

```

4885 030652 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
030654 030664          .WORD 40$          ;LOOP ON ERROR START LOCATION
030656 060030          .WORD 1$-40$+57764      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
030660 000000          .WORD 0              ;LOOP ON ERROR END LOCATION
030662 060056          .WORD 25$-40$+57764   ;DISABLE CACHE
030664 012737 001015 177746 40$: MOV #OFF,CCR      ;LOCATE TEST CODE TO LOW CACHE SPACE
030672 004437 002342    JSR R4,RELCTL    ;ADDRESS OF START OF NEXT TEST
030676 031076          .WORD 10$+2

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE

```

```

4886 030700 005037 177752          CLR CHR          ;LOAD AMR<8:0> WITH ALL 0'S
4887 030704 012705 070000          MOV #70000,R5   ;ADDRESS 70000 TO R5
4888 030710 005025 2$: CLR (R5)+    ;CLEAR ALL HI CACHE MAIN MEMORY
4889 030712 020527 100000          CMP R5,#100000
4890 030716 001374          BNE 2$
4891 030720 012705 070000          MOV #70000,R5   ;1ST ADDRESS LOCATION IN R5
4892 030724 012703 050000          MOV #50000,R3   ;ADDRESS 50000 IN R3
4893 030730 112737 000003 177750 1$: MOVB #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY
4894          ;STORE BITS TO BE WRITTEN TO
4895          ;CMR<11:9> ONLY DURING THE DESTINATION
4896          ;ACCESS OF AN INSTRUCTION.
4897          ;TDAR ALLOWS TAG PARITY STORE GENERATOR
4898          ;INPUTS TO BE LOADED FROM AMR<8:0>
4899 030736 012737 000015 177746    MOV #15,CCR     ;NO UCB SO AS TO WRITE ENABLE PARITY
4900          ;STORE
4901 030744 005713          TST (R3)
4902 030746 005715          TST (R5)
4903          ;WRITE 0'S INTO DATA/TAG PARITY STORE
4904 030750 005715          TST (R5)        ;ADDRESS LOCATION SPECIFIED BY R5
4905 030752 013701 177750          MOV CMR,R1     ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
4906 030756 000240 25$: NOP          ;SAVE CMR DATA
                                ;INSTRUCTION 'JMP 1$' PLACED HERE
                                ;FOR LOOP ON ERROR
4907 030762 105037 177750          CLRB CMR      ;DISABLE MAINT. MODE
4908 030766 012737 001015 177746    MOV #OFF,CCR   ;DISABLE CACHE
4909 030774 010537 047334          MOV R5,CA121  ;GET PARITY ADDRESS LOCATION USED
4910 031000 006237 047334          ASR CA121     ;PREPARE CA121 FOR TYPEOUT
4911 031004 032701 004000          BIT #HPB,R1   ;CHECK 0 HI BYTE PARITY STORE
4912 031010 001404          BEQ 7$       ;PASS

```

```

4913 031012 104406          ERROR          ;ERROR
          031014 031012          .WORD    .-2          ;-----
4914          ;DATA/TAG PARITY GEN. & STORE TESTS
4915          ;READING CACHE MAINT. REGISTER
4916          ;BIT 11 FOR UPPER BYTE PARITY DATA DID
4917          ;NOT RESULT IN 0.
4918 031016 047334          CA121          ;PRINT PARITY STORE ADDRESS LOCATION
4919          ;USED: CA<12:1>
4920 031020 000000          .WORD    0
4921 031022 032701 002000    7$: BIT    #LPB,R1          ;CHECK 0 FOR LOWER BYTE PARITY DATA
4922 031026 001404          BEQ     8$          ;PASS
4923 031030 104406          ERROR          ;ERROR
          ;-----
          031032 031030          .WORD    .-2
4924          ;DATA/TAG PARITY GEN. $ STORE TESTS
4925          ;READING CACHE MAINT. REGISTER
4926          ;BIT 10 FOR LOWER BYTE PARITY DATA DID
4927          ;NOT RESULT IN 0.
4928 031034 047334          CA121
4929 031036 000000          .WORD    0
4930 031040 032701 001000    8$: BIT    #TPB,R1          ;CHECK 0 FOR TAG PARITY DATA
4931 031044 001404          BEQ     9$          ;PASS
4932 031046 104406          ERROR          ;ERROR
          ;-----
          031050 031046          .WORD    .-2
4933          ;DATA/TAG PARITY GEN. AND STORAGE TESTS
4934          ;READING CACHE MAINT.REGISTER BIT 9 FOR
4935          ;TAG PARITY DATA DID NOT RESULT IN 0.
4936 031052 047334          CA121          ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
4937 031054 000000          .WORD    0
4938 031056 062705 000002    9$: ADD    #2,R5          ;NEXT PARITY STORE ADDRESS LOCATION
4939 031062 062703 000002    ADD    #2,R3
4940 031066 022705 100000    CMP    #100000,R5          ;HAVE ALL HI CACHE PARITY STORE ADDRESS
4941          ;LOCATIONS BEEN DONE
4942 031072 001316          BNE    1$          ;NO,CONTINUE
4943 031074 000240          NOP
          031076 005237 001472    10$: INC    $TESTN          ;END OF TEST
          ;INCREMENT TEST COUNTER

```

.SBTTL TEST # 163 - CHK SETTING HI CACHE DATA & TAG PARITY STORES  
\*\*\*\*\*  
\*TEST 163 - CHK SETTING HI CACHE DATA & TAG PARITY STORES  
\*WRITE AND READ 1'S TO ALL LOW CACHE DATA PARITY AND TAG PARITY STORES  
\*CONDITIONS:  
\* INPUTS TO DATA PARITY GEN:  
\* WRTD<15:0>= 000401  
\* WWPD(1)= 0  
\* INPUTS TO TAG PARITY GEN.:  
\* TAG WRTD<21:13> BIT PATTERN 00000001  
\* WWPT(1)=0  
\* DATA PARITY/TAG PARITY STORE ADDRESS:  
\* CA<12:1>=0000 TO 3777  
\* RESULT: CMR<11:9> ALL 1'S  
\*\*\*\*\*

4958 031102 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
031104 031114 .WORD 40\$ ;ERROR/LOOP ON TEST  
031106 070034 .WORD 1\$-40\$+67764 ;TEST START LOCATION  
031110 000000 .WORD 0 ;LOOP ON ERROR START LOCATION  
031112 070062 .WORD 25\$-40\$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
031114 012737 001015 177746 40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION  
031122 004437 002370 JSR R4,RELCTH ;DISABLE CACHE  
031126 031332 .WORD 10\$+2 ;LOCATE TEST CODE TO HIGH CACHE SPACE  
;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

4959 031130 012737 000001 177752 MOV #1,CHR ;LOAD AMR<8:0> WITH BIT PATTERN 00000001  
4960 031136 012705 060000 MOV #60000,R5 ;ADDRESS 60000 TO R5  
4961 031142 012725 000401 2\$: MOV #401,(R5)+ ;WRITE A 401 IN ALL LOW CACHE MAIN MEMORY  
4962 031146 020527 070000 CMP R5,#70000  
4963 031152 001373 BNE 2\$  
4964 031154 012705 060000 MOV #60000,R5 ;1ST ADDRESS LOCATION IN R5  
4965 031160 012703 040000 MOV #40000,R3 ;ADDRESS 40000 IN R3  
4966 031164 112737 000003 1\$: MOVB #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY  
4967 ; STORE BITS TO BE WRITTEN TO  
4968 ; CMR<11:9> ONLY DURING THE DESTINATION  
4969 ; ACCESS OF AN INSTRUCTION.  
4970 ; TDAR ALLOWS TAG PARITY STORE GENERATOR  
4971 ; INPUTS TO BE LOADED FROM AMR<8:0>  
4972 031172 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY STORE  
4973 031200 005713 TST (R3)  
4974 031202 005715 TST (R5)  
4975 ;WRITE 1'S INTO DATA/TAG PARITY STORE  
4976 031204 005715 TST (R5) ;ADDRESS LOCATION SPECIFIED BY R5  
4977 031206 013701 177750 MOV CMR,R1 ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>  
4978 031212 000240 25\$: NOP ;SAVE CMR DATA  
031214 000240 NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
4979 031216 105037 177750 CLRB CMR ;FOR LOOP ON ERROR  
4980 031222 012737 001015 177746 MOV #OFF,CCR ;DISABLE MAINT. MODE  
4981 031230 010537 047334 MOV R5,CA121 ;DISABLE CACHE  
4982 031234 006237 047334 MOV CA121 ;GET PARITY ADDRESS LOCATION USED  
4983 031240 032701 004000 ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
4984 031244 001004 BIT #HPB,R1 ;CHECK 1 HI BYTE PARITY STORE  
4985 031246 104406 BNE 7\$ ;PASS  
ERROR ;ERROR



```

031250 031246                    .WORD    .-2                    ;-----
4986                                                            ;DATA/TAG PARITY GEN. & STORE TESTS
4987                                                            ;READING CACHE MAINT. REGISTER
4988                                                            ;BIT 11 FOR UPPER BYTE PARITY DATA DID
4989                                                            ;NOT RESULT IN 1.
4990 031252 047334                CA121                        ;PRINT PARITY STORE ADDRESS LOCATION
4991                                                            ;USED: CA<12:1>
4992 031254 000000                .WORD    0                        ;CHECK 1 FOR LOWER BYTE PARITY DATA
4993 031256 032701 002000        7$:    BIT    #LPB,R1                    ;PASS
4994 031262 001004                BNE    8$                        ;ERROR
4995 031264 104406                ERROR                            ;-----

031266 031264                    .WORD    .-2                    ;DATA/TAG PARITY GEN. & STORE TESTS
4996                                                            ;READING CACHE MAINT. REGISTER
4997                                                            ;BIT 10 FOR LOWER BYTE PARITY DATA DID
4998                                                            ;NOT RESULT IN 1.
4999                                                            ;PRINT PARITY STORE ADDRESS USED:CA<12:1>
5000 031270 047334                CA121                        ;CHECK 1 FOR TAG PARITY DATA
5001 031272 000000                .WORD    0                        ;PASS
5002 031274 032701 001000        8$:    BIT    #TPB,R1                    ;ERROR
5003 031300 001004                BNE    9$                        ;-----
5004 031302 104406                ERROR                            ;DATA/TAG PARITY GEN. AND STORAGE TESTS
5005                                                            ;READING CACHE MAINT.REGISTER BIT 9 FOR
5006                                                            ;TAG PARITY DATA DID NOT RESULT IN 1.
5007                                                            ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
5008 031306 047334                CA121                        ;NEXT PARITY STORE ADDRESS LOCATION
5009 031310 000000                .WORD    0                        ;HAVE ALL LOW CACHE PARITY STORE ADDRESS
5010 031312 062705 000002        9$:    ADD    #2,R5                        ;LOCATIONS BEEN DONE
5011 031316 062703 000002        ADD    #2,R3                        ;NO CONTINUE
5012 031322 022705 070000        CMP    #70000,R5                    ;END OF TEST
5013                                                            ;INCREMENT TEST COUNTER
5014 031326 001316                BNE    1$                        ;
5015 031330 000240                NOP                                ;
031332 005237 001472                INC    $TESTN                    ;
  
```

\*\*\*\*\*  
:SBTTL TEST # 164 - CHK SETTING HI CACHE DATA & TAG PARITY STORES  
:\*\*\*\*\*  
\*TEST 164 - CHK SETTING HI CACHE DATA & TAG PARITY STORES  
\*WRITE AND READ 1'S TO ALL HI CACHE DATA PARITY AND TAG PARITY STORES  
\*CONDITIONS:  
\* INPUTS TO DATA PARITY GEN:  
\* WRD<15:0>= 000401  
\* WRPD(1)= 0  
\* INPUTS TO TAG PARITY GEN.:  
\* TAG WRD<21:13> =BIT PATTERN 00000001  
\* WPT(1)=0  
\* DATA PARITY/TAG PARITY STORE ADDRESS:  
\* CA<12:1>=4000 TO 7777  
\* RESULT: CMR<11:9> ALL 1'S  
:\*\*\*\*\*

5030 031336 000004  
031336 031350  
031340 031350  
031342 060034  
031344 000000  
031346 060062  
031350 012737 001015 177746 40\$:  
031356 004437 002342  
031362 031566

TST164:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
.WORD 40\$ ;LOOP ON ERROR START LOCATION  
.WORD 1\$-40\$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 0 ;LOOP ON ERROR END LOCATION  
.WORD 25\$-40\$+57764 ;DISABLE CACHE  
MOV #OFF,CCR ;LOCATE TEST CODE TO LOW CACHE SPACE  
JSR R4,RELCTL ;ADDRESS OF START OF NEXT TEST  
.WORD 10\$+2

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO LOW CACHE SPACE

5031 031364 012737 000001 177752  
5032 031372 012705 070000  
5033 031376 012725 000401 2\$:  
5034 031402 020527 100000  
5035 031406 001373  
5036 031410 012705 070000  
5037 031414 012703 050000  
5038 031420 112737 000003 177750 1\$:  
5039  
5040  
5041  
5042  
5043  
5044 031426 012737 000015 177746  
5045 031434 005713  
5046 031436 005715  
5047  
5048 031440 005715  
5049 031442 013701 177750  
5050 031446 000240 25\$:  
031450 000240  
5051 031452 105037 177750  
5052 031456 012737 001015 177746  
5053 031464 010537 047334  
5054 031470 006237 047334  
5055 031474 032701 004000  
5056 031500 001004  
5057 031502 104406

MOV #1,CHR ;LOAD AMR<8:0> WITH BIT PATTERN 00000001  
MOV #70000,R5 ;ADDRESS 70000 TO R5  
2\$: MOV #401,(R5)+ ;WRITE A 401 TO ALL HI CACHE  
CMP R5,#100000  
BNE 2\$  
MOV #70000,R5 ;1ST ADDRESS LOCATION IN R5  
MOV #50000,R3 ;ADDRESS 50000 IN R3  
1\$: MOV #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY  
; STORE BITS TO BE WRITTEN TO  
; CMR<11:9> ONLY DURING THE DESTINATION  
; ACCESS OF AN INSTRUCTION.  
; TDAR ALLOWS TAG PARITY STORE GENERATOR  
; INPUTS TO BE LOADED FROM AMR<8:0>  
; NO UCB SO AS TO WRITE ENABLE PARITY STORE  
MOV #15,CCR  
TST (R3)  
TST (R5) ;WRITE 1'S INTO DATA/TAG PARITY STORE  
; ADDRESS LOCATION SPECIFIED BY R5  
; WRITE DATA/TAG PARITY BITS INTO CMR<11:9>  
TST (R5)  
25\$: MOV CMR,R1 ;SAVE CMR DATA  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINT. MODE  
MOV #OFF,CCR ;DISABLE CACHE  
MOV R5,CA121 ;GET PARITY ADDRESS LOCATION USED  
ASR CA121 ;PREPARE CA121 FOR TYPEOUT  
BIT #HPB,R1 ;CHECK 1 HI BYTE PARITY STORE  
BNE 7\$ ;PASS  
ERROR ;ERROR

```

031504 031502 .WORD -2 ;-----
5058 ;DATA/TAG PARITY GEN. & STORE TESTS
5059 ;READING CACHE MAINT. REGISTER
5060 ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5061 ;NOT RESULT IN 1.
5062 031506 047334 CA121 ;PRINT PARITY STORE ADDRESS LOCATION
5063 ;USED: CA<12:1>
5064 031510 000000 .WORD 0
5065 031512 032701 002000 7$: BIT #LPB,R1 ;CHECK 1 FOR LOWER BYTE PARITY DATA
5066 031516 001004 BNE 8$ ;PASS
5067 031520 104406 ERROR ;ERROR
;-----
031522 031520 .WORD -2
5068 ;DATA/TAG PARITY GEN. & STORE TESTS
5069 ;READING CACHE MAINT. REGISTER
5070 ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5071 ;NOT RESULT IN 1.
5072 031524 047334 CA121 ;PRINT PARITY STORE ADDRESS USED:CA<12:1>
5073 031526 000000 .WORD 0
5074 031530 032701 001000 8$: BIT #TPB,R1 ;CHECK 1 FOR TAG PARITY DATA
5075 031534 001004 BNE 9$ ;PASS
5076 031536 104406 ERROR ;ERROR
;-----
031540 031536 .WORD -2
5077 ;DATA/TAG PARITY GEN. AND STORAGE TESTS
5078 ;READING CACHE MAINT.REGISTER BIT 9 FOR
5079 ;TAG PARITY DATA DID NOT RESULT IN 0.
5080 031542 047334 CA121 ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
5081 031544 000000 .WORD 0
5082 031546 062705 000002 9$: ADD #2,R5 ;NEXT PARITY STORE ADDRESS LOCATION
5083 031552 062703 000002 ADD #2,R3
5084 031556 022705 100000 CMP #100000,R5 ;HAVE ALL HI CACHE PARITY STORE ADDRESS
5085 ;LOCATIONS BEEN DONE
5086 031562 001716 BEQ 1$ ;NO,CONTINUE
5087 031564 000240 10$: NOP ;END OF TEST
031566 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```

```

SBTTL TEST # 165 - VERIFY BYTE DATA PARITY STORE ADDRESS LINES
*****
*TEST 165 - VERIFY BYTE DATA PARITY STORE ADDRESS LINES
*   VERIFY HI & LO BYTE DATA PARITY STORE ADDRESS LINES
*   PROCEDURE:   WRITE 0 INTO HI & LO BYTE DATA PARITY STORE
*                 ADDRESS LOCATION 0000.
*                 WRITE A 1 INTO HI & LO BYTE DATA PARITY STORE
*                 ADDRESS LOCAT. 0001.
*                 READ HI & LO BYTE DATA PARITY ADDRESS LOC.
*                 0000 FOR 0'S.
*                 REPEAT THE ABOVE SEQUENCE ,EACH TIME CHANGING THE
*                 ADDRESS LOCATION THE 1 IS WRITTEN INTO BY
*                 SHIFTING THE 1 ONE PLACE TO THE LEFT.
*****
  
```

```

TST165:
5101 031572 000004      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
                                40$
                                .WORD 40$
                                .WORD 1$-40$+67764
                                .WORD 0
                                .WORD 25$-40$+67764
031574 031604          .WORD 40$
031576 070040          .WORD 1$-40$+67764
031600 000000          .WORD 0
031602 070100          .WORD 25$-40$+67764
031604 012737 001015 177746 40$: MOV #OFF,CCR
031612 004437 002370 JSR R4,RELCTH
031616 032042          .WORD 10$+2

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

5102 031620 000240      NOP          ;THIS 'NOP' WILL BE AT LOCATION 70000
5103                                     ;WHEN THE TEST IS RELOCATED TO HI
5104                                     ;CACHE. IT WILL BE OVERWRITTEN WITH
5105                                     ;'401' WHEN THE TEST IS EXECUTED.
5106 031622 005037 060000 CLR 60000    ;CLEAR LOCATION 60000 IN MAIN MEMORY
5107 031626 012737 000002 047344 MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
5108 031634 012702 040000 2$: MOV #40000,R2 ;ADDRESS 40000 INTO R2
5109 031640 012703 060000 MOV #60000,R3 ;ADRESS 60000 INTO R3
5110 031644 063702 047344 ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
5111 031650 063703 047344 ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
5112 031654 012713 000401 MOV #401,(R3) ;ODD DATA IN HI & LO BYTE AREAS OF
5113                                     ;LOCATION SPECIFIED BY R3
5114 031660 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS HI & LO BYTE DATA PARITY
5115                                     ;STORE BITS TO BE WRITTEN TO CMR<11:10>
5116                                     ; ONLY DURING THE
5117                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
5118 031666 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE DATA PARITY STORE
5119 031674 005737 040000 TST 40000
5120 031700 005737 060000 TST 60000 ;READ UPDATE: WRITE 0'S INTO HI AND LO
5121                                     ;BYTE DATA PARITY STORES
5122 031704 005712 TST (R2)
5123 031706 005713 TST (R3) ;WRITE 1 INTO HI & LO BYTE DATA
5124                                     ;PARITY STORE LOCATION
5125                                     ;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
5126 031710 005737 060000 TST 60000 ;LOAD DATA FROM HI & LO BYTE PARITY
5127                                     ;DATA PARITY STORE LOCATION
5128                                     ;0000 INTO CMR<11:10>.
5129 031714 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
5130 031720 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
  
```

5131	031722	000240				NOP			:FOR LOOP ON ERROR
5132	031724	105037	177750			CLRB	CMR		:DISABLE MAINT. MODE
5133	031730	012737	001015	177746		MOV	#OFF,CCR		:DISABLE CACHE
5134	031736	032701	004000			BIT	#HPB,R1		:READING CMR<11> FOR HI BYTE
5135									:DATA PARITY STORE DATA SHOULD RESULT
5136	031742	001411				BEQ	8\$		:IN 0.
5137	031744	013737	047344	047334		MOV	FLTPAT,CA121		:PASS
5138	031752	006237	047334			ASR	CA121		:SAVE CA<12:1> USED
5139	031756	104406				ERROR			:PREPARE CA121 FOR TYPEOUT
									:ERROR
									:-----
	031760	031756				.WORD	.-2		
5140									:HI & LO BYTE DATA PARITY STORE ADDRESS TEST
5141									:HI BYTE DATA PARITY STORE LOC. 0000
5142									:DID NOT READ AS A 0 INDICATING THAT
5143									:IT WAS OVERWRITTEN WITH A 1. THIS
5144									:SUGGESTS HI BYTE DATA PARITY STORE
5145									:ADDRESS LINE IS BAD.
5146	031762	047334				CA121			:PRINT PARITY DTORE ADDRESS FAILURE
5147									:CA<12:1>.
5148									:NOTE THAT THE 1 IN THIS PATTERN
5149									:WILL POINT TO THE ADDRESS LINE OF
5150									:THAT BROUGHT OUT ERROR.
5151	031764	000000				.WORD	0		
5152	031766	032701	004000		8\$:	BIT	#HPB,R1		:READING CMR<10> FOR LO BYTE
5153									:DATA PARITY STORE DATA SHOULD RESULT
5154									:IN 0.
5155	031772	001411				BEQ	9\$		:PASS
5156	031774	013737	047344	047334		MOV	FLTPAT,CA121		:SAVE CA<12:1> USED
5157	032002	006237	047334			ASR	CA121		:PREPARE CA121 FOR TYPEOUT
5158	032006	104406				ERROR			:ERROR
									:-----
	032010	032006				.WORD	.-2		
5159									:HI & LO BYTE DATA PARITY STORE ADDRESS TEST
5160									:LO BYTE DATA PARITY STORE LOC. 0000
5161									:DID NOT READ AS A 0 INDICATING THAT
5162									:IT WAS OVERWRITTEN WITH A 1. THIS
5163									:SUGGESTS HI BYTE DATA PARITY STORE
5164									:ADDRESS LINE IS BAD.
5165	032012	047334				CA121			:PRINT PARITY STORE ADDRESS FAILURE
5166									:CA<12:1>.
5167									:NOTE THAT THE 1 IN THIS PATTERN
5168									:WILL POINT TO THE ADDRESS LINE OF
5169									:THAT BROUGHT OUT ERROR.
5170	032014	000000				.WORD	0		
5171	032016	006337	047344		9\$:	ASL	FLTPAT		:NEXT PATTERN
5172	032022	022737	020000	047344		CMP	#20000,FLTPAT		:HAS DATA PARITY STORE ADDRESS 4000 BEEN DONE?
5173	032030	001301				BNE	2\$		:NO
5174	032032	012737	000240	070000		MOV	#240,70000		:RESTORE OVERWRITTEN LOCATION 70000 WITH NOP.
5175	032040	000240			10\$:	NOP			:END OF TEST
	032042	005237	001472			INC	\$TESTN		:INCREMENT TEST COUNTER

CKKABO 11-44 KK11B CACHE  
 TEST # 166 - VERIFY TAG PARITY STORE ADDRESS LINES  
 5188

```

.SBTTL TEST # 166 - VERIFY TAG PARITY STORE ADDRESS LINES
*****
*TEST 166 - VERIFY TAG PARITY STORE ADDRESS LINES
*   VERIFY TAG PARITY STORE ADDRESS LINES
*   PROCEDURE:   WRITE 0 INTO TAG PARITY STORE
*               ADDRESS LOCATION 0000.
*               WRITE A 1 INTO TAG PARITY STORE
*               ADDRESS LOCAT. 0001.
*               READ TAG PARITY ADDRESS LOC.
*               0000 FOR 0 S.
*               REPEAT THE ABOVE SEQUENCE ,EACH TIME CHANGING THE
*               ADDRESS LOCATION THE 1 IS WRITTEN INTO BY
*               SHIFTING THE 1 ONE PLACE TO THE LEFT.
*****

```

```

5189 032046 000004
      032050 032060
      032052 070026
      032054 000000
      032056 070066
      032060 012737 001015 177746 40$:
      032066 004437 002370
      032072 032246

```

```

TST166:
      SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
            ;ERROR/LOOP ON TEST
            ;TEST START LOCATION
            ;LOOP ON ERROR START LOCATION
            ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
            ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR ;DISABLE CACHE
      JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

```

THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ARE RELOCATED TO HI CACHE SPACE

```

5190 032074 012737 000002 047344
5191 032102 012702 040000 2$:
5192 032106 012703 060000
5193 032112 063702 047344
5194 032116 063703 047344
5195 032122 112737 000002 177750 1$:
5196
5197
5198
5199 032130 012737 000015 177746
5200 032136 005737 040000
5201 032142 005737 060000
5202
5203 032146 005713
5204 032150 005712
5205
5206
5207 032152 005737 060000
5208
5209
5210 032156 013701 177750
5211 032162 000240 25$:
      032164 000240
5212 032166 105037 177750
5213 032172 012737 001015 177746
5214 032200 032701 001000
5215
5216 032204 001411
5217 032206 013737 047344 047334

```

```

      MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
      MOV #40000,R2 ;ADDRESS 40000 INTO R2
      MOV #60000,R3 ;ADRESS 60000 INTO R3
      ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
      ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
      MOVB #HODO,CMR ;HODO ALLOWS TAG PARITY
      ;STORE BITS TO BE WRITTEN TO CMR<9>
      ; ONLY DURING THE
      ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
      ;NO UCB SO AS TO WRITE ENABLE DATA PARITY STORE
      TST 40000
      TST 60000 ;READ UPDATE: WRITE 0 INTO TAG
      ; PARITY STORE
      TST (R3)
      TST (R2) ;WRITE 1 INTO TAG
      ;PARITY STORE LOCATION
      ;SPECIFIED BY R2'S BITS 1 THRU 12:CA<12:1>.
      TST 60000 ;LOAD DATA FROM
      ;TAG PARITY STORE LOCATION
      ;0000 INTO CMR<9>.
      MOV CMR,R1 ;SAVE CMR DATA
      NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP ;FOR LOOP ON ERROR
      CLRB CMR ;DISABLE MAINT. MODE
      MOV #OFF,CCR ;DISABLE CACHE
      BIT #TPB,R1 ;READING CMR<9> FOR TAG PARITY STORE
      ;DATA SHOULD RESULT IN 0
      BEQ 9$ ;PASS
      MOV FLTPAT,CA121 ;SAVE CA<12:1> USED

```

5218	032214	006237	047334				ASR	CA121		;PREPARE CA121 FOR TYPEOUT
5219	032220	104406					ERROR			;ERROR
	032222	032220					.WORD	.-2		;-----
5220										;TAG PARITY STORE ADDRESS TEST
5221										;TAG PARITY STORE LOC. 0000
5222										;DID NOT READ AS A 0 INDICATING THAT
5223										;IT WAS OVERWRITTEN WITH A 1. THIS
5224										;SUGGESTS TAG PARITY STORE
5225										;ADDRESS LINE IS BAD.
5226	032224	047334						CA121		;PRINT PARITY STORE ADDRESS FAILURE
5227										;CA<12:1>.
5228										;NOTE THAT THE 1 IN THIS PATTERN
5229										;WILL POINT TO THE ADDRESS LINE OF
5230										;THAT BROUGHT OUT ERROR.
5231	032226	000000					.WORD	0		
5232	032230	006337	047344		9\$:		ASL	FLTPAT		;NEXT PATTERN
5233	032234	022737	020000	047344			CMP	#20000,FLTPAT		;HAS TAG PARITY STORE ADDRESS 4000 BEEN DONE?
5234	032242	001317					BNE	2\$		;NO
5235	032244	000240			10\$:		NOP			;END OF TEST
	032246	005237	001472				INC	\$TESTN		;INCREMENT TEST COUNTER

5240

.SBTTL TEST # 167 - PARITY ERROR BITS IN CME=0 AFTER WRITE TO CME  
\*\*\*\*\*  
\*TEST 167 - PARITY ERROR BITS IN CME=0 AFTER WRITE TO CME  
\* VERIFY THAT ALL PARITY ERROR BITS IN CACHE MEMORY ERROR REGISTER  
\* WILL READ 0 FOLLOWING A WRITE TO CME.  
\*\*\*\*\*

5241 032252 000004  
032254 032264  
032256 070000  
032260 000000  
032262 070016  
032264 012737 001015 177746  
032272 004437 002370  
032276 032350

TST167:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

5242 032300 112737 000002 177750  
5243  
5244  
5245  
5246  
5247 032306 005037 177744  
5248 032312 013701 177744  
5249 032316 000240  
032320 000240  
5250 032322 105037 177750  
5251 032326 005701  
5252 032330 001406  
5253 032332 010137 047346  
5254 032336 104406  
032340 032336  
5255  
5256  
5257 032342 047346  
5258 032344 000000  
5259 032346 000240  
032350 005237 001472

1\$: MOVB #HODO,CMR ;HODO WILL ALLOW CLOCKING OF PARITY INFO.  
;TO CME ONLY DURING THE DESTINATION ACCESS OF AN  
;INSTRUCTION. THE EFFECT  
;IS THAT NO CLOCKING WILL OCCUR DURING EXECUTION  
;OF THE NEXT INSTRUCTION.  
CLR CME ;WRITE TO CME  
MOV CME,R1 ;SAVE CME CONTENTS.  
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLRB CMR ;DISABLE MAINTENANCE MODE  
TST R1 ;ARE ALL BITS 0?  
BEQ 10\$ ;PASS  
MOV R1,RECDAT ;GET CME CONTENTS RECEIVED  
ERROR ;ERROR  
;-----  
.WORD -2 ;PARITY ERROR CHECK TESTS  
;WRITING TO CME DID NOT LEAVE ALL PARITY ERROR BITS 0  
;PRINT CME CONTENTS RECEIVED  
10\$: .WORD 0  
NOP ;END OF TEST  
INC \$TESTN ;INCREMENT TEST COUNTER



TEST # 170 - CME CAN SHOW NO PARITY ERROR FOLLOWING READ HIT

5270

.SBTTL TEST # 170 - CME CAN SHOW NO PARITY ERROR FOLLOWING READ HIT

```

*****
*TEST 170 - CME CAN SHOW NO PARITY ERROR FOLLOWING READ HIT
*   VERIFY THAT CME CAN SHOW NO PARITY ERRORS FOLOWING A READ HIT CONDITION
*   VERIFY TAG/DATA 'PARITY CHECK PARITY GENERATORS' WITH ALL 0'S
*   ON THEIR INPUTS.
*   PROCEDURE:  CREATE ALL 0'S ON THE INPUTS OF THE TAG/DATA
*               PARITY CHECK PARITY GENERATORS DURING A READ
*               HIT CONDITION.ALLOW PARITY INFO. TO BE CLOCKED TO
*               CME.
*   RESULT:    CME<15>,<7>,<6>,<5> ALL 0'S
*****

```

TST170:

```

5271 032354 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
032356 032366          .WORD 40$          ;LOOP ON ERROR START LOCATION
032360 070016          .WORD 1$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
032362 000000          .WORD 0          ;LOOP ON ERROR END LOCATION
032364 070072          .WORD 25$-40$+67764 ;DISABLE CACHE
032366 012737 001015 177746 40$: MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE
032374 004437 002370     JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST
032400 032550          .WORD 10$+2

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

5272 032402 052737 000400 177746     BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5273 032410 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT TILL DONE
5274 032416 001374          BNE 200$
5275 032420 005037 000000          1$: CLR 0 ;0'S TO MAIN MEMORY LOCATION 0
5276 032424 112737 000002 177750     MOVB #HODO,CMR ;HODO ALLOWS CLOCKING OF PARITY INFO TO
5277          ;CME ONLY DURING THE DESTINATION ACCESS OF
5278          ;OF AN INSTRUCTION
5279 032432 012737 000015 177746     MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
5280 032440 005737 040000          TST 40000
5281 032444 005737 000000          TST 0 ;READ UPDATE TO CACHE LOCATION 0000-
5282          ;ALL 0'S WILL BE WRITTEN INTO DATA/TAG STORES
5283          ;AND A 0 INTO DATA/TAG PARITY STORES
5284 032450 005037 177744          CLR CME ;CLEAR CME
5285 032454 005737 000000          TST 0 ;READ HIT-ALL 0'S WILL BE PLACED ON INPUTS
5286          ;OF DATA/TAG PARITY DETECT PARITY GENERATORS
5287          ;AND ALL PARITY INFO WILL BE CLOCKED TO CME
5288 032460 052737 000200 177746     BIS #PEA,CCR ;SET CCR<7> SO AS TO ENABLE CME<7>,<6>,<5> TO
5289          ;TO BE WRITTEN INDIVIDUALLY FROM
5290          ;PARITY INFO LOGIC,AND TO WRITE CME<15>.
5291 032466 000240          NOP
5292 032470 013701 177744          MOV CME,R1 ;SAVE CME CONTENTS
5293 032474 000240          NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
5294 032476 000240          NOP ;FOR LOOP ON ERROR
5295 032500 012737 001015 177746     MOV #OFF,CCR ;DISABLE CACHE
5296 032506 105037 177750          CLRB CMP ;DISABLE MAINTENANCE MODE
5297 032512 012737 000002 000000     MOV #2,0 ;RESTORE LOCATION 0
5298 032520 005701          TST R1 ;WERE ALL PARITY ERROR BITS IN CME=0?
5299 032522 001411          BEQ 10$ ;PASS;NEXT TEST
5300 032524 005037 047332          CLR EXDAT6 ;SPECIFY CME CONTENTS EXPECTED
5301 032534 104406          MOV R1,RECDAT ;GET CME CONTENTS RECEIVED
                                ;ERROR

```

5302 032536 032534  
5303  
5304 032540 047332  
5305 032542 047346  
5306 032544 000000  
5307 032546 000240  
032550 005237 001472

.WORD .-2  
  
EXDAT6  
RECDAT  
10\$: .WORD 0  
NOP  
INC \$TESTN

;-----  
;PARITY CHECK TESTS  
;ALL PARITY ERROR BITS IN CME SHOULD HAVE READ 0  
;PRINT EXPECTED CME CONTENTS  
;PRINT CONTENTS OF CME RECEIVED  
  
;END OF TEST  
;INCREMENT TEST COUNTER

CKKKABO 11-44 KK11B CACHE  
TEST # 171 - PARITY ERROR BIT CHECK  
5331

MACRO M1113 27-OCT-80 08:39 PAGE 142 SEQUENCE 201

```

.SBTTL TEST # 171 - PARITY ERROR BIT CHECK
*****
*TEST 171 - PARITY ERROR BIT CHECK
*VERIFY THE FOLLOWING WHEN A LOCATION PREVIOUSLY WRITTEN
*WITH WRONG TAG PARITY IS ACCESSED:
*   1. A PARITY ERROR IS DETECTED AND ALL PARITY ERROR BITS IN
*      CME READ CORRECTLY WITH PEA CLEARED,
*   2. ALL PARITY ERROR ERROR BITS READ CORRECTLY WITH PEA
*      SET.
*   3. A WRITE TO CME CLEARS CME<15> AND <5> FROM A 1 STATE
PROCEDURE:   WRITE WRONG PARITY TO TAG PARITY STORE LOCATION
              0000. CLOCK PARITY INFO. TO CME.
CONDITIONS: DATA PARITY CHECK PARITY GEN. INPUTS:
              ALL 0'S
              TAG PARITY CHECK PAR. GEN. INPUTS:
              TAGD<20:13>= ALL 0'S
              TAG PARITY BIT=1
RESULTS:     PEA CLEARED:
              CME<15>=0
              CME<7>,<6>,<5>=1
              PEA SET:
              CME<15>=1
              CME<7>,<6>=0
              CME<5>=1
*****

```

```

5332 032554 000004
      032556 032566
      032560 070016
      032562 000000
      032564 070126
      032566 012737 001015 177746
      032574 004437 002370
      032600 033106

```

```

TST171:
      SCPCND           ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                      ;ERROR/LOOP ON TEST
                      ;TEST START LOCATION
      .WORD 40$        ;LOOP ON ERROR START LOCATION
      .WORD 1$-40$+67764
      .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR    ;DISABLE CACHE
      JSR R4,RELCTH  ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2     ;ADDRESS OF START OF NEXT TEST

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

```

```

5333 032602 052737 000400 177746 2$:  BIS #FC,CCR
5334 032610 032737 010000 177746 300$: BIT #VCIP,CCR
5335 032616 001374          BNE 300$
5336 032620 005037 000000 1$:  CLR 0
5337 032624 112737 000002 177750  MOVB #HODO,CMR
5338
5339
5340 032632 012737 000015 177746  MOV #15,CCR
5341 032640 005737 000000  TST 0
5342 032644 005737 040000  TST 40C00
5343 032650 052737 002000 177746  BIS #WWPT,CCR
5344 032656 005737 000000  TST 0
5345
5346
5347
5348 032662 042737 002000 177746  BIC #WWPT,CCR
5349 032670 005037 177744  CLR CME
5350 032674 005737 000000  TST 0

```

```

;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
;WAIT TILL DONE
;O'S TO MAIN MEMORY LOCATION 0.
;HODO ALLOWS CLOCKING OF PARITY INFO TO
;CME ONLY DURING THE DESTINATION ACCESS OF
;AN INSTRUCTION.
;NO UCB SO AS TO WRITE CACHE STORES
;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
;ALLOW WRITE WRONG PARITY TO TAGG PARITY STORE
;READ UPDATE TO CACHE LOCATION 0000;
;ALL 0'S(EVEN DATA) WILL BE WRITTEN TO DATA/TAG STORES.
;O'S INTO DATA PARITY STORES,AND A 1
;INTO TAG PARITY STORE.
;DISABLE WWPT
;CLEAR CME
;READ HIT; ALL 0'S WILL BE PLACED ON INPUTS

```



5401  
5402 033076 047332  
5403 033100 047346  
5404 033102 000000  
5405 033104 000240  
033106 005237 001472

10\$:

EXDAT6  
RECDAT  
.WORD 0  
NOP  
INC \$TESTN

:FOLLOWING WRITE TO CME  
:PRINT CME CONTENTS EXPECTED  
:PRINT CME CONTENTS RECEIVED  
:END OF TEST  
:INCREMENT TEST COUNTER

TEST # 172 - VERIFY WRONG BYTE INFO CAUSES PROPER PARITY ERROR

5432

.SBTTL TEST # 172 - VERIFY WRONG BYTE INFO CAUSES PROPER PARITY ERROR

\*\*\*\*\*

\*TEST 172 - VERIFY WRONG BYTE INFO CAUSES PROPER PARITY ERROR

\*VERIFY THE FOLLOWING WHEN A LOCATION PREVIOUSLY WRITTEN  
\*WITH WRONG LO & HI BYTE PARITY IS ACCESSED.

- \*1. A PARITY ERROR IS DETECTED AND ALL PARITY ERROR BITS IN CME READ CORRECTLY WITH PEA CLEARED,
- \*2. ALL PARITY ERROR BITS READ CORRECTLY WITH PEA SET.
- \*3. A WRITE TO CME CLEARS CME<15> ,<7> AND <6> FROM A 1 STATE

PROCEDURE: WRITE WRONG PARITY TO LO BYTE PARITY STORE LOCATION 0000. CLOCK PARITY INFO. TO CME.

CONDITIONS: HI BYTE PARITY CHECK PARITY GEN. INPUTS:

INTD<15:8>=0

HI BYTE PARITY BIT=1

LO BYTE PAR. CHECK PAR. GEN INPUTS:

INTD<7:0>= ALL 0'S

LO BYTE PARITY BIT=1

TAG PARITY CHECK PARITY GEN. INPUTS:

ALL 0'S

RESULTS: PEA CLEARED:

CME<15>=0

CME<7>,<6>,<5>=1

PEA SET:

CME<15>=1

CME<7>,<6>=1

CME<5>=0

\*\*\*\*\*

TST172:

5433 033112 000004

SCPCND

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON

:ERROR/LOOP ON TEST

:TEST START LOCATION

:LOOP ON ERROR START LOCATION

:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST

:IJOB ON ERROR END LOCATION

:DISABLE CACHE

:LOCATE TEST CODE TO HIGH CACHE SPACE

:ADDRESS OF START OF NEXT TEST

033114 033124  
 033116 070016  
 033120 000000  
 033122 070126  
 033124 012737  
 033132 004437  
 033136 033444

001015 177746 40\$:  
002370

.WORD 40\$  
 .WORD 1\$-40\$+67764  
 .WORD 0  
 .WORD 25\$-40\$+67764  
 MOV #OFF,CCR  
 JSR R4,RELCTH  
 .WORD 10\$+2

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

5434 033140 052737 000400 177746  
 5435 033146 032737 010000 177746  
 5436 033154 001374  
 5437 033156 005037 000000  
 5438 033162 112737 000002 177750  
 5439  
 5440  
 5441 033170 012737 000015 177746  
 5442 033176 005737 000000  
 5443 033202 005737 040000  
 5444 033206 052737 000100 177746  
 5445  
 5446 033214 005737 000000  
 5447  
 5448

200\$:  
 1\$:  
 BIS #FC,CCR  
 BIT #VCIP,CCR  
 BNE 200\$  
 CLR 0  
 MOVB #HODO,CMR  
 MOV #15,CCR  
 TST 0  
 TST 40000  
 BIS #WWPD,CCR  
 TST 0

:FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS

:WAIT TILL DONE

:0'S TO MAIN MEMORY LOCATION 0.

:HODO ALLOWS CLOCKING OF PARITY INFO TO

:CME ONLY DURING THE DESTINATION ACCESS OF

:AN INSTRUCTION.

:NO UCB SO AS TO WRITE CACHE STORES

:UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE

:ALLOW WRITE WRONG PARITY DATA TO LO

:& HI BYTE PARITY STORE.

:READ UPDATE TO CACHE LOCATION 0000;

:ALL 0'S WILL BE WRITTEN TO DATA/TAG STORES,

:1'S INTO LO & HI BYTE DATA PARITY STORES,AND A 0







TEST # 173 - INT. LOGIC TRAPS ACCESSING LOC WITH BAD PARITY  
5517

.SBTTL TEST # 173 - INT. LOGIC TRAPS ACCESSING LOC WITH BAD PARITY  
\*\*\*\*\*  
\*TEST 173 - INT. LOGIC TRAPS ACCESSING LOC WITH BAD PARITY  
\* VERIFY INTERRUPT LOGIC BY ASSURING THAT A TRAP OCCURS TO LOCATION  
\* 114 WHEN A LOCATION PREVIOUSLY WRITTEN  
\* WITH WRONG HI/LO BYTE PARITY IS ACCESSED.  
\* CONDITIONS: PEA=0  
\* DCPI=0  
\*\*\*\*\*

5518 033450 000004  
033452 033462  
033454 070016  
033456 000000  
033460 070152  
033462 012737 001015 177746 40\$:  
033470 004437 002370  
033474 033734

TST173:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR P4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

5519 033476 052737 000400 177746  
5520 033504 032737 010000 177746 200\$:  
5521 033512 001374  
5522 033514 005037 000000 1\$:  
5523 033520 012737 070142 000114  
5524 033526 012737 000340 000116  
5525 033534 112737 000002 177750  
5526  
5527  
5528 033542 005037 002070  
5529 033546 012737 000015 177746  
5530 033554 005737 000000  
5531 033560 005737 040000  
5532 033564 052737 000100 177746  
5533  
5534 033572 005737 000000  
5535  
5536 033576 042737 000100 177746  
5537 033604 005037 177744  
5538 033610 042737 000005 177746  
5539  
5540 033616 005737 000000  
5541  
5542  
5543  
5544  
5545 033622 000240  
5546 033624 005237 002070  
5547 033630 012737 001015 177746  
5548 033636 000404  
5549 033640 012737 001015 177746 4\$:  
5550 033646 022626  
5551 033650 000240 25\$:  
033652 000240

BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS  
BIT #VCIP,CCR ;WAIT TILL DONE  
BNE 200\$  
CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.  
MOV #4\$-40\$+67764,114 ;SETUP FOR CACHE TRAP  
MOV #340,116  
MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES  
;AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC  
; ONLY DURING THE DESTINATION ACCESS OF AN INSTRUCTION.  
CLR FAIL1 ;CLEAR ERROR FLAG  
MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES  
TST 0  
TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE  
BIS #WYPD,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO  
; & HI BYTE PARITY STORE.  
TST 0 ;READ UPDATE TO CACHE LOCATION 0000;  
;WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES  
BIC #WYPD,CCR ;DISABLE WYPD  
CLR CME ;CLEAR CME AND PARITY DETECT LOGIC  
BIC #DCPI+FMLO,CCR ;ALLOW FOR INTERRUPT TO OCCUR  
;AND ENABLE LOW CACHE  
TST 0 ;READ HIT;  
;LO & HI BYTE PARITY CHECK GENERATORS WILL  
; DETECT WRONG PARITY AND THE PARITY  
;ERROR WILL BE CLOCKED TO INTERRUPT  
;LOGIC  
NOP  
INC FAIL1 ;INDICATE THAT TRAP DID NOT OCCUR  
MOV #OFF,CCR ;DISABLE CACHE  
BR 25\$  
MOV #OFF,CCR ;DISABLE CACHE  
CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR

5552	033654	105037	177750			CLRB	CMR	:DISABLE MAINT. MODE
5553	033660	012737	000002	000000		MOV	#2,0	:RESTORE LOCATION 0
5554	033666	012737	000116	000114		MOV	#116,114	:RESTORE CACHE INTERRUPT VECTORS
5555	033674	005037	000116			CLR	116	
5556	033700	052737	000400	177746		BIS	#FC,CCR	:BEFORE LEAVING TEST FLUSH CACHE TO
5557								:ELIMINATE ANY EFFECTS OF WWPD
5558	033706	032737	010000	177746	500\$:	BIT	#VCIP,CCR	:WAIT TILL DONE
5559	033714	001374				BNE	SC0\$	
5560	033716	005737	002070			TST	FAIL1	:DID TRAP OCCUR?
5561	033722	001403				BEQ	10\$	:YES
5562	033724	104406				ERROR		:ERROR
								:-----
	033726	033724				.WORD	.-2	
5563								:INTERRUPT/ABORT LOGIC TESTS
5564								:TRAP TO LOCATION 114 DID NOT OCCUR
5565	033730	000000				.WORD	0	
5566	033732	000240			10\$:	NOP		:END OF TEST
	033734	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

.SBTTL TEST # 174 - VERIFY INTERRUPT LOGIC TRAP CAN BE INHIBITED

\*\*\*\*\*
:TEST 174 - VERIFY INTERRUPT LOGIC TRAP CAN BE INHIBITED
:VERIFY INTERRUPT LOGIC BY ASSURING THAT A TRAP CAN BE INHIBITED TO LOCATION
:114 WHEN A LOCATION PREVIOUSLY WRITTEN
:WITH WRONG HI/LO BYTE PARITY IS ACCESSED.
:CONDITIONS: PEA=0
:DCPI=1
\*\*\*\*\*

5575 033740 000004

033742 033752
033744 070016
033746 000000
033750 070152
033752 012737
033760 004437
033764 034224

001015 177746
002370

TST174:

SCPCND

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
:DISABLE CACHE
:LOCATE TEST CODE TO HIGH CACHE SPACE
:ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$
:ARE RELOCATED TO HI CACHE SPACE

5576 033766 052737 000400 177746
5577 033774 032737 010000 177746
5578 034002 001374
5579 034004 005037 000000
5580 034010 012737 070136 000114
5581 034016 012737 000340 000116
5582 034024 112737 000002 177750
5583
5584
5585
5586 034032 005037 002070
5587 034036 012737 000015 177746
5588 034044 005737 000000
5589 034050 005737 040000
5590 034054 052737 000100 177746
5591
5592 034062 005737 000000
5593
5594 034066 042737 000100 177746
5595 034074 005037 177744
5596 034100 042737 000004 177746
5597 034106 005737 000000
5598
5599
5600
5601
5602 034112 000240
5603 034114 012737 001015 177746
5604 034122 000406
5605 034124 012737 001015 177746
5606 034132 005237 002070
5607 034136 022626
5608 034140 000240
034142 000240

200\$:

BIS #FC,CCR
BIT #VCIP,CCR
BNE 200\$
CLR 0
MOV #4\$-40\$+67764,114
MOV #340,116
MOV #HODO,CMR

:FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
:WAIT TILL DONE

1\$:

CLR 0
MOV #4\$-40\$+67764,114
MOV #340,116
MOV #HODO,CMR

:0'S TO MAIN MEMORY LOCATION 0.
:SETUP FOR CACHE TRAP

:HODO ALLOWS CACHE UPDATES
:AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC
: ONLY DURING THE DESTINATION ACCESS OF
:AN INSTRUCTION.
:CLEAR ERROR FLAG
:NO UCB SO AS TO WRITE CACHE STORES
:UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
:ALLOW WRITE WRONG PARITY DATA TO LO
:& HI BYTE PARITY STORE.
:READ UPDATE TO CACHE LOCATION 0000;
:WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES
:DISABLE WWPDP
:CLEAR CME AND PARITY DETECT LOGIC
:ENABLE LO CACHE
:READ HIT;
:LO & HI BYTE PARITY CHECK GENERATORS WILL
:DETECT WRONG PARITY AND THE PARITY
:ERROR WILL BE CLOCKED TO INTERRUPT
:LOGIC

4\$:

NOP
MOV #OFF,CCR
BR 25\$
MOV #OFF,CCR
INC FAIL1
CMP (SP)+,(SP)+

:DISABLE CACHE
:DISABLE CACHE
:INDICATE THAT TRAP OCCURED
:READJUST STACK DUE TO INTERRUPT
:INSTRUCTION 'JMP 1\$' PLACED HERE
:FOR LOOP ON ERROR

25\$:

NOP
NOP

5609	034144	105037	177750			CLRB	CMR		:DISABLE MAINT. MODE
5610	034150	012737	000002	000000		MOV	#2,0		:RESTORE LOCATION 0
5611	034156	012737	000116	000114		MOV	#116,114		:RESTORE CACHE INTERRUPT VECTORS
5612	034164	005037	000116			CLR	116		
5613	034170	052737	000400	177746		BIS	#FC,CCR		:BEFORE LEAVING TEST FLUSH CACHE TO
5614									:ELIMINATE ANY EFFECTS OF WWPD
5615	034176	032737	010000	177746	500\$:	BIT	#VCIP,CCR		:WAIT TILL DONE
5616	034204	001374				BNE	500\$		
5617	034206	005737	002070			TST	FAIL1		:DID TRAP OCCUR?
5618	034212	001403				BEQ	10\$		:NO
5619	034214	104406				ERROR			:ERROR
									:-----
	034216	034214				.WORD	.-2		
5620									: INTERRUPT/ABORT LOGIC TESTS
5621									: TRAP TO LOCATION 114 OCCURED
5622									: AND WAS NOT INHIBITED BY DCPI
5623	034220	000000				.WORD	0		
5624	034222	000240			10\$:	NOP			:END OF TEST
	034224	005237	001472			INC	\$TESTN		:INCREMENT TEST COUNTER

SBTTL TEST # 175 - CME BIT 15 OPERATES PROPERLY & TRAP TO 114 OCCURS  
\*\*\*\*\*  
\*TEST 175 - CME BIT 15 OPERATES PROPERLY & TRAP TO 114 OCCURS  
\* VERIFY ABORT LOGIC BY THE FOLLOWING RESULTS WHEN A LOCATION  
\* PREVIOUSLY WRITTEN WITH WRONG HI/LO BYTE PARITY IS ACCESSED.  
\* 1. CME<15> WILL SET CAUSED BY ABORT SIGNAL BEING ASSERTED  
\* 2. WRITE TO CME WILL CLEAR CME<15>  
\* 3. INSTRUCTION CYCLE WILL BE ABORTED  
\* 4. THE ABORT CAUSES TRAP TO 114  
\* PROCEDURE: INHIBIT CLOCKING OF PARITY ERROR SIGNAL TO  
\* INTERRUPT LOGIC. ALLOW CME<15> TO BE SET  
\* BY ABORT SIGNAL WHICH IS ASSERTED BY PARITY  
\* ERROR SIGNAL TO ABORT LOGIC.  
\* CONDITIONS: PEA=1  
\* DCPI=1  
\*\*\*\*\*

TST175:  
5640 034230 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
034232 034242 .WORD 40\$ ;TEST START LOCATION  
034234 070016 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
034236 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
034240 070174 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
034242 012737 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE  
034250 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
034254 034602 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST  
  
;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE  
  
5641 034256 052737 000400 177746 BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS  
5642 034264 037737 010000 177746 200\$: BIT #VCIP,CCR ;WAIT TILL DONE  
5643 034272 001374 BNE 200\$  
5644 034274 005037 000000 1\$: CLR 0 ;ALL 0'S TO LOCATION 0  
5645 034300 005000 CLR R0 ;ADDRESS 0 TO R0  
5646 034302 012737 070150 000114 MOV #4\$-40\$+67764,114 ;SETUP FOR TRAP  
5647 034310 012737 000340 000116 MOV #340,116  
5648 034316 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES  
5649 ;AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC  
5650 ; ONLY DURING THE DESTINATION ACCESS OF  
5651 ;AN INSTRUCTION.  
5652 034324 005037 002070 CLR FAIL1 ;CLEAR ERROR FLAG  
5653 034330 012703 177777 MOV #-1,R3 ;ALL 1'S TO R3  
5654 034334 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES  
5655 034342 005710 TST (R0)  
5656 034344 005737 040000 TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE  
5657 034350 052737 000100 177746 BIS #WVPD,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO  
5658 ; & HI BYTE PARITY STORE.  
5659 034356 005710 TST (R0) ;READ UPDATE TO CACHE LOCATION 0000  
5660 ;WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES  
5661 034360 042737 000100 177746 BIC #WVPD,CCR ;DISABLE WVPD  
5662 034366 005037 177744 CLR CME ;CLEAR CME AND PARITY DETECT LOGIC  
5663 034372 042737 000004 177746 BIC #FMLO,CCR ;ENABLE LOW CACHE  
5664 034400 052737 000200 177746 BIS #PEA,CCR ;ALLOW FOR ABORT  
5665 034406 011003 MOV (R0),R3 ;READ HIT;  
5666 ;LO & HI BYTE PARITY CHECK GENERATORS WILL  
5667 ; DETECT WRONG PARITY

```

5668                                     ;USING HODO AND SOURCE MODE FOR READING
5669                                     ;LOCATION 0 WILL INHIBIT PARITY ERROR
5670                                     ;FROM BEING CLOCKED TO INTERRUPT LOGIC
5671                                     ;HOWEVER, THE PARITY ERROR SIGNAL
5672                                     ;WILL CAUSE THE ABORT SIGNAL TO BE
5673                                     ;ASSERTED. THE ABORT SIGNAL WILL BE
5674                                     ;CAUSE CME<15> TO BE SET.
5675                                     ;THIS INSTRUCTION SHOULD BE ABORTED
5676 034410 000240                       NOP
5677 034412 005237 002070                 INC     FAIL1
5678 034416 012737 001015 177746         MOV     #OFF,CCR
5679 034424 000404                       BR      5$
5680 034426 012737 001015 177746 4$:    MOV     #OFF,CCR
5681 034434 022626                       CMP     (SP)+,(SP)+
5682 034436 013701 177744                 MOV     CME,R1
5683 034442 005037 177744                 CLR     CME
5684 034446 013702 177744                 MOV     CME,R2
5685 034452 000240                       NOP
5686 034454 000240                       NOP
5687 034456 105037 177750                 CLRB   CMR
5688 034462 012737 000002 000000         MOV     #2,0
5689 034470 012737 000116 000114         MOV     #116,114
5690 034476 005037 000116                 CLR     116
5691 034502 052737 000400 177746         BIS     #FC,CCR
5692 034510 032737 010000 177746 6$:    BIT     #VCIP,CCR
5693 034516 001374                       BNE    6$
5694 034520 032701 100000                 BIT     #CMPE,R1
5695 034524 001003                       BNE    7$
5696 034526 104406                       ERROR
5697 034530 034526                       .WORD  -2
5698                                     ;INTERRUPT/ABORT LOGIC
5699                                     ;CME<15> WAS NOT SET DUE TO ABORT
5700                                     ;SIGNAL
5701 034532 000000                       .WORD  0
5702 034534 032702 100000 7$:          BIT     #CMPE,R2
5703 034540 001403                       BEQ    8$
5704 034542 104406                       ERROR
5705 034544 034542                       .WORD  -2
5706                                     ;INTERRUPT/ABORT TESTS
5707                                     ;CME<15> WAS NOT CLEARED BY WRITE TO
5708                                     ; TO CME
5709 034546 000000                       .WORD  0
5710 034550 022703 177777 8$:          CMP     #-1,R3
5711 034554 001403                       BEQ    9$
5712 034556 104406                       ERROR
5713 034560 034556                       .WORD  -2
5714                                     ;INTERRUPT/ABORT TESTS
5715                                     ;R3 WAS OVERWRITTEN WITH DATA INDICATING
5716                                     ;THAT INSTRUCTION WAS NOT ABORTED
5717 034562 000000                       .WORD  0
5718 034564 005737 002070 9$:          TST     FAIL1
5719 034570 001403                       BEQ    10$
5720 034572 104406                       ERROR
    
```

```
5718 034574 034572 .WORD -2 ;-----  
5719 ;INTERRUPT/ABORT TESTS  
5720 034576 000000 .WORD 0 ;TRAP DID NOT OCCUR DUE TO ABORT  
5721 034600 000240 10$: NOP ;END OF TEST  
034602 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
```

.SBTTL TEST # 176 - CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S  
\*\*\*\*\*  
\*TEST 176 - CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S  
\*VERIFY CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S BY PERFORMING A  
\*MARCH PATTERN TEST TO LOW CACHE AREA OF TAG/DATA PARITY STORE(LOC. 0000-3777)  
\*PROCEDURE: 1. WRITE ALL 0'S TO ALL LO CACHE TAG/DATA PARITY STORE  
\*RAMS CORRESPONDING TO LOCATIONS 0000-3777  
\*2. READ 0'S FROM ALL LO CACHE RAMS CORRESPONDING  
\*TO LOCATION 0000  
\*3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION  
\*0000.  
\*4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION  
\*0000.  
\*5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION  
\*AND UNTIL LOC. 3777 IS REACHED.  
\*\*\*\*\*

5737 034606 000004  
034610 034620  
034612 070000  
034614 000000  
034616 070126  
034620 012737 001015 177746  
034626 004437 002370  
034632 035234

TST176:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

5738 034634 012702 000401  
5739 034640 005037 177752  
5740 034644 012700 060000  
5741 034650 005020  
5742 034652 020027 070000  
5743 034656 001374  
5744 034660 012700 060000  
5745 034664 012701 040000  
5746 034670 112737 000003 177750  
5747  
5748  
5749  
5750  
5751  
5752 034676 012737 000015 177746  
5753 034704 005721  
5754 034706 005720  
5755  
5756 034710 022700 070000  
5757 034714 001373  
5758 034716 012737 000001 177752  
5759 034724 012700 060000  
5760 034730 042737 000001 177750  
5761  
5762 034736 005710  
5763  
5764

1\$: MOV #401,R2 ;SETUP R2 WITH PATTERN 401  
CLR CHR ;LOAD AMR<8:0> WITH ALL 0'S  
MOV #60000,R0 ;ADDRESS 60000 TO R0  
2\$: CLR (R0)+ ;CLEAR ALL LOW CACHE MAIN MEMORY  
CMP R0,#70000  
BNE 2\$  
MOV #60000,R0 ;1ST ADDRESS LOCATION IN R0  
MOV #40000,R1 ;ADDRESS 40000 IN R1  
MOV #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & DATA /TAG PARITY  
; STORE BITS TO BE WRITTEN TO  
;CMR<11:9> ONLY DURING THE DESTINATION  
;ACCESS OF AN INSTRUCTION.  
;TDAR ALLOWS TAG PARITY STORE GENERATOR  
;INPUTS TO BE LOADED FROM AMR<8:0>  
;NO UCB SO AS TO WRITE ENABLE PARITY STORE  
6\$: MOV #15,CCR  
TST (R1)+  
TST (R0)+ ;WRITE 0'S INTO ALL DATA/TAG PARITY STORE  
;ADDRESS LOCATIONS SPECIFIED BY R0  
;DONE?  
;NO  
MOV #1,CHR ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>  
MOV #60000,R0 ;ADDR. 60000 TO R0  
7\$: BIC #TDAR,CMR ;DISABLE TDAR TO ALLOW UPDATE  
;OF CACHE TAG STORE THRU CA<21:13>  
TST (R0) ;READ MISS TO CACHE LOCATION SPECIFIED  
;BY R0. CLOCK TAG/DATA PARITY STORE  
;BITS INTO CMR<11:9>.SHOULD BE ALL 0'S.



```

5765 ;ALSO CAUSES UPDATE TO CACHE.TAG/DATA
5766 ;PARITY STORE LOCATION SHOULD REMAIN
5767 ;WITH 0'S.
5768 034740 013705 177750 MOV CMR,R5 ;SAVE CMR CONTENTS
5769 034744 052737 000001 177750 BIS #TDAR,CMR ;ENABLE TDAR TO ALLOW TAG PARITY GENERATOR
5770 ;INPUTS TO SEE ODD DATA FROM AMR<8:0>
5771 034752 010210 MOV R2,(R0) ;WRITE HIT CAUSES UPDATE TO CACHE.
5772 ;TAG/DATA PARITY STORES WILL BE WRITTEN
5773 ;WITH 1'S DUE TO AMR<8:0> ODD DATA
5774 ;AND PATTERN 401 FROM R2 BEING PUT
5775 ;ONTO PAX DATA LINES RESULTING IN
5776 ;ODD DATA FOR LO AND HI BYTE DATA PARITY GENERATORS.
5777 034754 005710 TST (R0) ;READ MISS.CLOCK TAG/DATA PARITY STORE BITS TO
5778 ;CMR<11:9>.SHOULD BE ALL 1'S.
5779 034756 013703 177750 MOV CMR,R3 ;SAVE CMR CONTENTS
5780 034762 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
5781 034766 042705 170777 BIC #170777,R5 ;FOR LOOP ON ERROR
5782 034772 005705 TST R5 ;INTERESTED IN BITS 11:9
5783 034774 001437 BEQ 8$ ;BITS 11:9 SHOULD BE ALL 0'S
5784 034776 012737 001015 177746 MOV #OFF,CCR ;PASS
5785 035004 105037 177750 CLRB CMR ;DISABLE CACHE
5786 035010 005037 047336 CLR EXDAT1 ;CLEAR MAINT. MODE
5787 035014 010537 047354 MOV R5,CMR119 ;SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
5788 ;SPECIFY CACHE TAG/DATA PARITY STORE DATA RECEIVED
5789 035020 012737 000011 002062 MOV #9,LOOP ;THRU CMR<11:9>
5790 035026 006237 047354 11$: ASR CMR119 ;PREPARE CMR119 FOR TYPEOUT
5791 035032 042737 100000 047354 BIC #100000,CMR119
5792 035040 005337 002062 DEC LOOP
5793 035044 001370 BNE 11$
5794 035046 010037 047334 MOV RO,CA121 ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
5795 035052 006237 047334 ASR CA121
5796 035056 104406 ERROR ;ERROR
5797 035060 035056 .WORD .-2 ;-----
5798 ;TAG/DATA PARITY STORE MARCH PATTERN TEST
5799 ;READING CACHE TAG/DATA PARITY STORE DATA
5800 ;THRU CMR<11:9> DID NOT READ ALL 0'S.
5801 ;THIS SUGGESTS THAT A RAM LOCATION
5802 ;SPECIFIED BY CA121 WAS OVERWRITTEN
5803 ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
5804 ;LOCATION.ANY BIT IN CMR119 DATA
5805 ;THAT IS A 1 MAY POINT TO A BAD
5806 ;CACHE TAG/DATA PARITY STORE RAM.
5807 035062 047336 EXDAT1 ;PRINT EXPECTED CACHE TAG/DATA PARITY STORE DATA
5808 035064 047354 CMR119 ;PRINT CACHE TAG/DATA PARITY STORE RECEIVED
5809 ;THRU CMR<11:9>
5810 035066 047334 CA121 ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
5811 035070 000000 .WORD 0
5812 035072 000452 BR 3$ ;END THE TEST
5813 035074 042703 170777 8$: BIC #170777,R3 ;INTERESTED IN BITS 11:9 ONLY
5814 035100 022703 007000 CMP #7000,R3 ;BITS 11:9 SHOULD BE ALL 1'S
5815 035104 001440 BEQ 9$ ;PASS
5816 035106 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
5817 035114 105037 177750 CLRB CMR ;CLEAR MAINT. MODE
5818 035120 012737 000007 047336 MOV #7,EXDAT1 ;SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
    
```



SBTTL TEST # 177 - CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S  
 \*\*\*\*\*  
 \*TEST 177 - CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S  
 \* VERIFY CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S BY PERFORMING A  
 \* MARCH PATTERN TEST TO HIGH CACHE AREA OF TAG/DATA PARITY STORE (LOC. 4000-7777)  
 \* PROCEDURE: 1. WRITE ALL 0'S TO ALL HI CACHE TAG/DATA PARITY STORE  
 \* RAMS CORRESPONDING TO LOCATIONS 4000-7777  
 \* 2. READ 0'S FROM ALL HI CACHE RAMS CORRESPONDING  
 \* TO LOCATION 4000  
 \* 3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION  
 \* 4000  
 \* 4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION  
 \* 4000  
 \* 5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION  
 \* AND UNTIL LOC. 7777 IS REACHED.  
 \*\*\*\*\*

```

TST177:
5862 035240 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
035242 035252          .WORD 40$          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
035244 060000          .WORD 1$-40$+57764 ;LOOP ON ERROR END LOCATION
035246 000000          .WORD 0           ;DISABLE CACHE
035250 060126          .WORD 25$-40$+57764 ;LOCATE TEST CODE TO LOW CACHE SPACE
035252 012737 001015 177746 40$: MOV #OFF,CCR
035260 004437 002342     JSR R4,RELCTL
035264 035666          .WORD 10$+2       ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE

5863 035266 012702 000401 1$: MOV #401,R2          ;SETUP R2 WITH PATTERN 401
5864 035272 005037 177752     CLR CHR          ;LOAD AMR<8:0> WITH ALL 0'S
5865 035276 012700 070000     MOV #70000,R0   ;ADDRESS 70000 TO R0
5866 035302 005020          2$: CLR (R0)+    ;CLEAR ALL HIGH CACHE MAIN MEMORY
5867 035304 020027 100000     CMP R0,#100000
5868 035310 001374          BNE 2$
5869 035312 012700 070000     MOV #70000,R0   ;1ST ADDRESS LOCATION IN R0
5870 035316 012701 050000     MOV #50000,R1   ;ADDRESS 50000 IN R1
5871 035322 112737 000003 177750 MOVB #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & DATA /TAG PARITY
5872                                ; STORE BITS TO BE WRITTEN TO
5873                                ;CMR<11:9> ONLY DURING THE DESTINATION
5874                                ;ACCESS OF AN INSTRUCTION.
5875                                ;TDAR ALLOWS TAG PARITY STORE GENERATOR
5876                                ;INPUTS TO BE LOADED FROM AMR<8:0>
5877 035330 012737 000015 177746 6$: MOV #15,CCR      ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
5878 035336 005721          TST (R1)+
5879 035340 005720          TST (R0)+
5880                                ;WRITE 0'S INTO ALL DATA/TAG PARITY STORE
5881 035342 022700 100000     CMP #100000,R0  ;ADDRESS LOCATIONS SPECIFIED BY R0
5882 035346 001373          BNE 6$         ;DONE?
5883 035350 012737 000001 177752 7$: MOV #1,CHR      ;NO
5884 035356 012700 070000     MOV #70000,R0   ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
5885 035362 042737 000001 177750 7$: BIC #TDAR,CMR  ;ADDR. 70000 TO R0
5886                                ;DISABLE TDAR TO ALLOW UPDATE
5887 035370 005710          TST (R0)       ;OF CACHE TAG STORE THRU CA<21:13>
5888                                ;READ MISS TO CACHE LOCATION SPECIFIED
5889                                ;BY R0. CLOCK TAG/DATA PARITY STORE
                                ;BITS INTO CMR<11:9>.SHOULD BE ALL 0'S.
  
```

```

5890
5891
5892
5893 035372 013705 177750      MOV    CMR,R5
5894 035376 052737 000001 177750  BIS    #TDAR,CMR
5895
5896 035404 010210      MOV    R2,(R0)
5897
5898
5899
5900
5901
5902
5903 035406 005710      TST    (R0)
5904
5905 035410 013703 177750      MOV    CMR,R3
5906 035414 000240      25$:  NGP
      035416 000240      NOP
5907 035420 042705 170777      BIC    #170777,R5
5908 035424 005705      TST    R5
5909 035426 001437      BEQ    8$
5910 035430 012737 001015 177746  MOV    #OFF,CCR
5911 035436 105037 177750      CLR    CMR
5912 035442 005037 047336      CLR    EXDAT1
5913 035446 010537 047354      MOV    R5,CMR119
5914
5915 035452 012737 000011 002062  MOV    #9,LOOP
5916 035460 006237 047354      11$:  ASR    CMR119
5917 035464 042737 100000 047354  BIC    #100000,CMR119
5918 035472 005337 002062      DEC    LOOP
5919 035476 001370      BNE    11$
5920 035500 010037 047334      MOV    R0,CA121
5921 035504 006237 047334      ASR    CA121
5922 035510 104406      ERROR
      035512 035510      .WORD  .-2
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933 035514 047336      EXDAT1
5934 035516 047354      CMR119
5935
5936 035520 047334      CA121
5937 035522 000000      .WORD  0
5938 035524 000452      BR     3$
5939 035526 042703 170777      8$:  BIC    #170777,R3
5940 035532 022703 007000      CMP    #7000,R3
5941 035536 001440      BEQ    9$
5942 035540 012737 001015 177746  MOV    #OFF,CCR
5943 035546 105037 177750      CLR    CMR
    
```

```

;ALSO CAUSES UPDATE TO CACHE.TAG/DATA
;PARITY STORE LOCATION SHOULD REMAIN
;WITH 0'S.
;SAVE CMR CONTENTS
;ENABLE TDAR TO ALLOW TAG PARITY GENERATOR
;INPUTS TO SEE ODD DATA FROM AMR<8:0>
;WRITE HIT CAUSES UPDATE TO CACHE.
;TAG/DATA PARITY STORES WILL BE WRITTEN
;WITH 1'S DUE TO AMR<8:0> ODD DATA
;AND PATTERN 401 FROM R2 BEING PUT
;ONTO PAX DATA LINES RESULTING IN
;ODD DATA FOR LO AND HI BYTE DATA PARITY
;GENERATORS.
;READ MISS.CLOCK TAG/DATA PARITY STORE BITS TO
;CMR<11:9>.SHOULD BE ALL 1'S.
;SAVE CMR CONTENTS
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;INTERESTED IN BITS 11:9
;BITS 11:9 SHOULD BE ALL 0'S
;PASS
;DISABLE CACHE
;CLEAR MAINT. MODE
;SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
;SPECIFY CACHE TAG/DATA PARITY STORE DATA RECEIVED
;THRU CMR<11:9>
;PREPARE CMR119 FOR TYPEOUT
;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
;ERROR
;-----
;TAG/DATA PARITY STORE MARCH PATTERN TEST
;READING CACHE TAG/DATA PARITY STORE DATA
;THRU CMR<11:9> DID NOT READ ALL 0'S.
;THIS SUGGESTS THAT A RAM LOCATION
;SPECIFIED BY CA121 WAS OVERWRITTEN
;WITH A 1 WHEN WRITING A 1 TO ANOTHER
;LOCATION.ANY BIT IN CMR119 DATA
;THAT IS A 1 MAY POINT TO A BAD
;CACHE TAG/DATA PARITY STORE RAM.
;PRINT EXPECTED CACHE TAG/DATA PARITY STORE DATA
;PRINT CACHE TAG/DATA PARITY STORE RECEIVED
;THRU CMR<11:9>
;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
;END THE TEST
;INTERESTED IN BITS 11:9 ONLY
;BITS 11:9 SHOULD BE ALL 1'S
;PASS
;DISABLE CACHE
;CLEAR MAINT. MODE
    
```



CKKABO 11-44 KK11B CACHE  
TEST # 200 - VERIFY CACHE VALID  
5988

MACRO M1113 27-OCT-80 08:39 PAGE 149 SEQUENCE 220

```

.SBTTL TEST # 200 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
*TEST 200 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
* MARCH PATTERN TEST TO LOW CACHE AREA OF VALID STORE(LOC. 0000-3777)
* PROCEDURE: 1. WRITE 0'S TO LO CACHE VALID STORE
*              RAM CORRESPONDING TO LOCATIONS 0000-3777
*              2. READ 0 FROM LO CACHE RAM CORRESPONDING
*                 TO LOCATION 0000
*              3. WRITE 1 TO RAM CORRESPONDING TO LOCATION
*                 0000.
*              4. READ 1 FROM RAM CORRESPONDING TO LOCATION
*                 0000.
*              5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*                 AND UNTIL LOC. 3777 IS REACHED.
*****

```

```

5989 035672 000004
      035674 035704
      035676 070026
      035700 000000
      035702 070136
      035704 012737 001015 177746
      035712 004437 002370
      035716 036222

```

```

TST200:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
      .WORD 40$             ;TEST START LOCATION
      .WORD 1$-40$+67764   ;LOOP ON ERROR START LOCATION
      .WORD 0               ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR          ;DISABLE CACHE
      JSR R4,RELCTH        ;LOCATE TEST CODE TO HIGH CACHE SPACE
      .WORD 10$+2          ;ADDRESS OF START OF NEXT TEST

```

```

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

```

```

5990 035720 032737 020000 177746
5991 035726 001407
5992 035730 052737 000400 177746
5993 035736 032737 010000 177746
5994 035744 001374
5995 035746 012700 060000
5996 035752 012701 040000
5997 035756 005002
5998 035760 112737 000002 177750
5999
6000
6001
6002
6003 035766 012737 000015 177746
6004 035774 005721
6005 035776 005720
6006
6007 036000 022700 070000
6008 036004 001373
6009 036006 012700 060000
6010 036012 052737 001000 177746
6011 036020 010220
6012
6013 036022 022700 070000
6014 036026 001374
6015 036030 042737 001000 177746
6016 036036 012700 060000

```

```

      BIT #VSIU,CCR        ;IS SET A USED
      BEQ 1$               ;YES
      BIS #FC,CCR         ;FLUSH
      BIT #VCIP,CCR
      BNE 200$
      MOV #60000,R0        ;1ST ADDRESS LOCATION IN R0
      MOV #40000,R1        ;ADDRESS 40000 IN R1
      CLR R2
      MOVB #HODO,CMR      ;HODO ALLOWS CACHE UPDATES & VALID
                          ;STORE BITS TO BE WRITTEN TO
                          ;CMR<12> ONLY DURING THE DESTINATION
                          ;ACCESS OF AN INSTRUCTION.
                          ;STORE
      MOV #15,CCR          ;NO UCB SO AS TO UPDATE VALID STORE
      TST (R1)+
      TST (R0)+
                          ;UPDATE ALL LO CACHE VALID STORE
                          ;ADDRESS LOCATIONS SPECIFIED BY R0
      CMP #70000,R0        ;DONE?
      BNE 6$
      MOV #60000,R0        ;ADDR. 60000 TO R0
      BIS #UCB,CCR         ;ENABLE UCB
      MOV R2,(R0)+        ;WRITE HIT WITH UCB WILL INVALIDATE
                          ;OR WRITE 0 TO ALL LO CACHE VALID STORE
      CMP #70000,R0        ;DONE?
      BNE 13$
      BIC #UCB,CCR         ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
      MOV #60000,R0        ;ADDRESS 60000 TO R0

```

TEST #	ADDRESS	DATA	VALID STORE	RAM MEMORY IC'S	INSTR	OPERANDS	COMMENTS
6017	036042	005710			7\$: TST	(R0)	:READ MISS TO CACHE LOCATION SPECIFIED
6018							:BY R0. CLOCK VALID STORE
6019							:BIT INTO CMR<12> SHOULD BE 0.
6020							:ALSO CAUSES UPDATE TO CACHE.
6021							:VALID STORE LOCATION WILL BE WRITTEN WITH A 1.
6022	036044	013705	177750		MOV	CMR,R5	:SAVE CMR CONTENTS
6023	036050	005710			TST	(R0)	:READ HIT.CLOCK VALID STORE BIT TO CMR<12> SHOULD BE 1.
6024	036052	013703	177750		MOV	CMR,R3	:SAVE CMR CONTENTS
6025	036056	000240			25\$: NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
6026	036062	042705	167777		NOP		:FOR LOOP ON ERROR
6027	036066	005705			BIC	#167777,R5	:INTERESTED IN BIT 12
6028	036070	001416			TST	R5	:BIT 12 SHOULD BE 0
6029	036072	012737	001015	177746	BEQ	8\$	:PASS
6030	036100	105037	177750		MOV	#OFF,CCR	:DISABLE CACHE
6031	036104	010037	047334		CLRB	CMR	:CLEAR MAINT. MODE
6032	036110	006237	047334		MOV	R0,CA121	:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6033	036114	104406			ASR	CA121	
					ERROR		:ERROR
	036116	036114			.WORD	.-2	:-----
6034							:VALID STORE MARCH PATTERN TEST- SET A
6035							:READING CACHE VALID STORE DATA
6036							:THRU CMR<12> DID NOT READ 0.
6037							:THIS SUGGESTS THAT A RAM LOCATION
6038							:SPECIFIED BY CA121 WAS OVERWRITTEN
6039							:WITH A 1 WHEN WRITING A 1 TO ANOTHER
6040							:LOCATION.
6041							:THIS INDICATES THAT VALID STORE RAM
6042							:SET A IS BAD.
6043							
6044	036120	047334			CA121		:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6045	036122	000000			.WORD	0	
6046	036124	000430			BR	3\$	:END THE TEST
6047	036126	042703	167777		8\$: BIC	#167777,R3	:INTERESTED IN BIT 12 ONLY
6048	036132	022703	010000		CMP	#10000,R3	:BIT 12 SHOULD BE 1
6049	036136	001416			BEQ	9\$	:PASS
6050	036140	012737	001015	177746	MOV	#OFF,CCR	:DISABLE CACHE
6051	036146	105037	177750		CLRB	CMR	:CLEAR MAINT. MODE
6052	036152	010037	047334		MOV	R0,CA121	:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6053	036156	006237	047334		ASR	CA121	
6054	036162	104406			ERROR		:ERROR
							:-----
	036164	036162			.WORD	.-2	
6055							:VALID STORE MARCH PATTERN TEST- SET A
6056							:READING CACHE VALID STORE DATA
6057							:THRU CMR<12> DID NOT READ 1.
6058							:THIS SUGGESTS THAT VALID STORE RAM
6059							:IC SET A IS BAD.
6060							:THRU CMR<12>
6061	036166	047334			CA121		:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6062	036170	000000			.WORD	0	
6063	036172	000405			BR	3\$	:END TEST
6064	036174	062700	000002		9\$: ADD	#2,R0	:NEXT LOCATION
6065	036200	022700	070000		CMP	#70000,R0	:HAS ALL LO CACHE BEEN DONE?
6066	036204	001316			BNE	7\$	:NO,CONTINUE
6067	036206	012737	001015	177746	3\$: MOV	#OFF,CCR	:DISABLE CACHE
6068	036214	105037	177750		CLRB	CMR	:DISABLE MAINT. MODE

6069 036220 000240  
036222 005237 001472

10\$:    NOP  
         INC    \$TESTN

:END OF TEST  
:INCREMENT TEST COUNTER



CKKABO 11-44 KK11B CACHE  
TEST # 201 - VERIFY CACHE VALID  
6084

MACRO M1113 27-OCT-80 08:39 PAGE 150 SEQUENCE 223  
STORE RAM MEMORY IC'S

.SBTTL TEST # 201 - VERIFY CACHE VALID STORE RAM MEMORY IC'S  
\*\*\*\*\*  
\*TEST 201 - VERIFY CACHE VALID STORE RAM MEMORY IC'S  
\*VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A  
\*MARCH PATTERN TEST TO HIGH CACHE AREA OF VALID STORE(LOC. 4000-7777)  
\*PROCEDURE: 1. WRITE 0'S TO HI CACHE VALID STORE  
\*RAM CORRESPONDING TO LOCATIONS 4000-7777  
\*2. READ 0 FROM HI CACHE RAM CORRESPONDING  
\*TO LOCATION 4000  
\*3. WRITE 1 TO RAM CORRESPONDING TO LOCATION  
\*4000  
\*4. READ 1 FROM RAM CORRESPONDING TO LOCATION  
\*4000  
\*5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION  
\*AND UNTIL LOC. 7777 IS REACHED.  
\*\*\*\*\*

6085 036226 000004  
036230 036240  
036232 060026  
036234 000000  
036236 060136  
036240 012737 001015 177746  
036246 004437 002342  
036252 036556

TST201:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
.WORD 40\$ ;LOOP ON ERROR START LOCATION  
.WORD 1\$-40\$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 0 ;LOOP ON ERROR END LOCATION  
.WORD 25\$-40\$+57764 ;DISABLE CACHE  
MOV #OFF,CCR ;LOCATE TEST CODE TO LOW CACHE SPACE  
JSR R4,RELCTL ;ADDRESS OF START OF NEXT TEST  
.WORD 10\$+2

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO LOW CACHE SPACE

6086 036254 032737 020000 177746  
6087 036262 001407  
6088 036264 052737 000400 177746  
6089 036272 032737 010000 177746  
6090 036300 001374  
5091 036302 012700 070000  
6092 036306 012701 050000  
6093 036312 005002  
6094 036314 112737 000002 177750  
6095  
6096  
6097  
6098 036322 012737 000015 177746  
6099 036330 005721  
6100 036332 005720  
6101  
6102 036334 022700 100000  
6103 036340 001373  
6104 036342 012700 070000  
6105 036346 052737 001000 177746  
6106 036354 010220  
6107  
6108 036356 022700 100000  
6109 036362 001374  
6110 036364 042737 001000 177746  
6111 036372 012700 070000  
6112 036376 005710

BIT #VSIU,CCR ;IS SET A USED  
BEQ 1\$ ;YES  
BIS #FC,CCR ;FLUSH  
200\$: BIT #VCIP,CCR  
BNE 200\$  
1\$: MOV #70000,R0 ;1ST ADDRESS LOCATION IN R0  
MOV #50000,R1 ;ADDRESS 50000 IN R1  
CLR R2  
MOV #HODO,CMR ;HODO ALLOWS CACHE UPDATES & VALID  
;STORE BITS TO BE WRITTEN TO  
;CMR<12> ONLY DURING THE DESTINATION  
;ACCESS OF AN INSTRUCTION. STORE  
;NO UCB SO AS TO UPDATE VALID STORE  
6\$: MOV #15,CCR  
TST (R1)+  
TST (R0)+ ;UPDATE ALL HI CACHE VALID STORE  
;ADDRESS LOCATIONS SPECIFIED BY R0  
;DONE?  
;NO  
;ADDR. 70000 TO R0  
;ENABLE UCB  
13\$: MOV #100000,R0  
BNE 6\$  
MOV #70000,R0  
BIS #UCB,CCR  
MOV R2,(R0)+ ;WRITE HIT WITH UCB WILL INVALIDATE  
;OR WRITE 0 TO ALL HI CACHE VALID STORE  
;DONE?  
;NO  
;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES  
;ADDRESS 70000 TO R0  
7\$: MOV #70000,R0  
TST (R0) ;READ MISS TO CACHE LOCATION SPECIFIED

```

TEST # 201 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
6113
6114
6115
6116
6117 036400 013705 177750 MOV CMR,R5
6118 036404 005710 TST (R0)
6119
6120 036406 013703 177750 MOV CMR,R3
6121 036412 000240 25$: NOP
        036414 000240 NOP
6122 036416 042705 167777 BIC #167777,R5
6123 036422 005705 TST R5
6124 036424 001416 BEQ 8$
6125 036426 012737 001015 177746 MOV #OFF,CCR
6126 036434 105037 177750 CLRБ CMR
6127 036440 010037 047334 MOV R0,CA121
6128 036444 006237 047334 ASR CA121
6129 036450 104406 ERROR
        036452 036450 .WORD -2
6130
6131
6132
6133
6134
6135
6136
6137
6138 036454 047334 CA121
6139 036456 000000 .WORD 0
6140 036460 000430 BR 3$
6141 036462 042703 167777 8$: BIC #167777,R3
6142 036466 022703 010000 CMP #10000,R3
6143 036472 001416 BEQ 9$
6144 036474 012737 001015 177746 MOV #OFF,CCR
6145 036502 105037 177750 CLRБ CMR
6146 036506 010037 047334 MOV R0,CA121
6147 036512 006237 047334 ASR CA121
6148 036516 104406 ERROR
        036520 036516 .WORD -2
6149
6150
6151
6152
6153
6154
6155 036522 047334 CA121
6156 036524 000000 .WORD 0
6157 036526 000405 BR 3$
6158 036530 062700 000002 9$: ADD #2,R0
6159 036534 022700 100000 CMP #100000,R0
6160 036540 001316 BNE 7$
6161 036542 012737 001015 177746 3$: MOV #OFF,CCR
6162 036550 105037 177750 CLRБ CMR
6163 036554 000240 10$: NOP
        036556 005237 001472 INC $TESTN

```

```

:BY R0. CLOCK VALID STORE
:BIT INTO CMR<12>. SHOULD BE 0.
:ALSO CAUSES UPDATE TO CACHE.
:VALID STORE LOCATION WILL BE WRITTEN WITH A 1.
:SAVE CMR CONTENTS
:READ HIT.CLOCK VALID STORE BIT TO CMR<12>
:SHOULD BE 1.
:SAVE CMR CONTENTS
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:INTERESTED IN BIT 12
:BIT 12 SHOULD BE 0
:PASS
:DISABLE CACHE
:CLEAR MAINT. MODE
:SPECIFY FAILED VALID STORE ADDRESS LOCATION
:ERROR
:-----
:VALID STORE MARCH PATTERN TEST- SET A
:READING CACHE VALID STORE DATA
:THRU CMR<12> DID NOT READ 0.
:THIS SUGGESTS THAT A RAM LOCATION
:SPECIFIED BY CA121 WAS OVERWRITTEN
:WITH A 1 WHEN WRITING A 1 TO ANOTHER
:LOCATION.
:THIS INDICATES THAT VALID STORE RAM SET A IS BAD.
:SPECIFY FAILED VALID STORE ADDRESS LOCATION
:END THE TEST
:INTERESTED IN BIT 12 ONLY
:BIT 12 SHOULD BE 1
:PASS
:DISABLE CACHE
:CLEAR MAINT. MODE
:SPECIFY FAILED VALID STORE ADDRESS LOCATION
:ERROR
:-----
:VALID STORE MARCH PATTERN TEST- SET A
:READING CACHE VALID STORE DATA
:THRU CMR<12> DID NOT READ 1.
:THIS SUGGESTS THAT VALID STORE RAM
:IC SET A IS BAD.
:THRU CMR<12>
:SPECIFY FAILED VALID STORE ADDRESS LOCATION
:END TEST
:NEXT LOCATION
:HAS ALL HI CACHE BEEN DONE?
:NO,CONTINUE
:DISABLE CACHE
:DISABLE MAINT. MODE
:END OF TEST
:INCREMENT TEST COUNTER

```

```

.SBTTL TEST # 202 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
*TEST 202 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
*MARCH PATTERN TEST TO LOW CACHE AREA OF VALID STORE(LOC. 0000-3777)
*PROCEDURE: 1. WRITE 0'S TO LO CACHE VALID STORE
*RAM CORRESPONDING TO LOCATIONS 0000-3777
*2. READ 0 FROM LO CACHE RAM CORRESPONDING
*TO LOCATION 0000
*3. WRITE 1 TO RAM CORRESPONDING TO LOCATION
*0000.
*4. READ 1 FROM RAM CORRESPONDING TO LOCATION
*0000.
*5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*AND UNTIL LOC. 3777 IS REACHED.
*****
  
```

```

6179 036562 000004      TST202:      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
036564 036574          .WORD 40$      ;TEST START LOCATION
036566 070026          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
036570 000000          .WORD 0        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
036572 070136          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
036574 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
036602 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
036606 037112          .WORD 10$+2    ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

6180 036610 032737 020000 177746 BIT #VSIU,CCR ;IS SET B USED
6181 036616 001007 BNE 1$ ;YES
6182 036620 052737 000400 177746 BIS #FC,CCR ;FLUSH
6183 036626 032737 010000 177746 200$: BIT #VCIP,CCR
6184 036634 001374 BNE 200$
6185 036636 012700 060000 1$: MOV #60000,R0 ;1ST ADDRESS LOCATION IN R0
6186 036642 012701 040000 MOV #40000,R1 ;ADDRESS 40000 IN R1
6187 036646 005007 CLR R2
6188 036650 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & VALID
6189 ; STORE BITS TO BE WRITTEN TO
6190 ;CMR<12> ONLY DURING THE DESTINATION
6191 ;ACCESS OF AN INSTRUCTION. STORE
6192 036656 012737 000015 177746 6$: MOV #15,CCR ;NO UCB SO AS TO UPDATE VALID STORE
6193 036664 005721 TST (R1)+
6194 036666 005720 TST (R0)+ ;UPDATE ALL LO CACHE VALID STORE
6195 ;ADDRESS LOCATIONS SPECIFIED BY R0
6196 036670 022700 070000 CMP #70000,R0 ;DONE?
6197 036674 001373 BNE 6$ ;NO
6198 036676 012700 060000 MOV #60000,R0 ;ADDR. 60000 TO R0
6199 036702 052737 001000 177746 BIS #UCB,CCR ;ENABLE UCB
6200 036710 010220 13$: MOV R2,(R0)+ ;WRITE HIT WITH UCB WILL INVALIDATE
6201 ;OR WRITE 0 TO ALL LO CACHE VALID STORE
6202 036712 022700 070000 CMP #70000,R0 ;DONE?
6203 036716 001374 BNE 13$ ;NO
6204 036720 042737 001000 177746 BIC #UCB,CCR ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
6205 036726 012700 060000 MOV #60000,R0 ;ADDRESS 60000 TO R0
6206 036732 005710 7$: TST (R0) ;READ MISS TO CACHE LOCATION SPECIFIED
  
```



6259 037104 105037 177750  
6260 037110 000240  
      037112 005237 001472

10\$:      CLR    CLR    CLR  
          NOP    NOP    NOP  
          INC    INC    INC

          CMR                    ;DISABLE MAINT. MODE  
          \$TESTN                ;END OF TEST  
                                 ;INCREMENT TEST COUNTER

```

.SBTTL TEST # 203 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
*TEST 203 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
* MARCH PATTERN TEST TO HIGH CACHE AREA OF VALID STORE(LOC. 4000-7777)
* PROCEDURE: 1. WRITE 0'S TO HI CACHE VALID STORE
*              RAM CORRESPONDING TO LOCATIONS 4000-7777
*              2. READ 0 FROM HI CACHE RAM CORRESPONDING
*              TO LOCATION 4000
*              3. WRITE 1 TO RAM CORRESPONDING TO LOCATION
*              4000
*              4. READ 1 FROM RAM CORRESPONDING TO LOCATION
*              4000
*              5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*              AND UNTIL LOC. 77 IS REACHED.
*****
    
```

```

TST203:
6276 037116 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          .WORD 40$          ;ERROR/LOOP ON TEST
          .WORD 1$-40$+57764 ;TEST START LOCATION
          .WORD 0           ;LOOP ON ERROR START LOCATION
          .WORD 25$-40$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD 10$+2       ;LOOP ON ERROR END LOCATION
          MOV #OFF,CCR      ;DISABLE CACHE
          JSR R4,RELCTL     ;LOCATE TEST CODE TO LOW CACHE SPACE
          .WORD 10$+2       ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE

6277 037144 032737 020000 177746      BIT #VSIU,CCR ;IS SET B USED
6278 037152 001007          BNE 1$        ;YES
6279 037154 052737 000400 177746      BIS #FC,CCR  ;FLUSH
6280 037162 032737 010000 177746      200$: BIT #VCIP,CCR
6281 037170 001374          BNE 200$
6282 037172 012700 070000          1$: MOV #70000,R0 ;1ST ADDRESS LOCATION IN R0
6283 037176 012701 050000          MOV #50000,R1 ;ADDRESS 50000 IN R1
6284 037202 005002          CLR R2
6285 037204 112737 000002 177750      MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & VALID
6286          ; STORE BITS TO BE WRITTEN TO
6287          ;CMR<12> ONLY DURING THE DESTINATION
6288          ;ACCESS OF AN INSTRUCTION.
6289          ;STORE
6290 037212 012737 000015 177746      6$: MOV #15,CCR ;NO ICB SO AS TO UPDATE VALID STORE
6291 037220 005721          TST (R1)+
6292 037222 005720          TST (R0)+
6293          ;UPDATE ALL HI CACHE VALID STORE
6294 037224 022700 100000          CMP #100000,R0 ;ADDRESS LOCATIONS SPECIFIED BY R0
6295 037230 001373          BNE 6$        ;DONE?
6296 037232 012700 070000          MOV #70000,R0 ;NO
6297 037236 052737 001000 177746      13$: BIS #UCB,CCR ;ADDR. 70000 TO R0
6298 037244 010220          MOV R2,(R0)+ ;ENABLE UCB
6299          ;WRITE HIT WITH UCB WILL INVALIDATE
6300 037246 022700 100000          CMP #100000,R0 ;OR WRITE 0 TO ALL HI CACHE VALID STORE
6301 037252 001374          BNE 13$      ;DONE?
6302 037254 042737 001000 177746      BIC #UCB,CCR ;NO
6303 037262 012700 070000          MOV #70000,R0 ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
          ;ADDRESS 70000 TO R0
    
```

```

6304 037266 005710          7$:   TST      (R0)          ;READ MISS TO CACHE LOCATION SPECIFIED
6305                                     ;BY R0.  CLOCK VALID STORE
6306                                     ;BIT INTO CMR<12>.SHOULD BE 0.
6307                                     ;ALSO CAUSES UPDATE TO CACHE.
6308                                     ;VALID STORE LOCATION WILL BE WRITTEN
6309                                     ;WITH A 1.
6310 037270 013705 177750      MOV      CMR,R5          ;SAVE CMR CONTENTS
6311 037274 005710          TST      (R0)          ;READ HIT.CLOCK VALID STORE BIT TO CMR<12>
6312                                     ;SHOULD BE 1.
6313 037276 013703 177750      MOV      CMR,R3          ;SAVE CMR CONTENTS
6314 037302 000240          25$:   NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
        037304 000240          NOP              ;FOR LOOP ON ERROR
6315 037306 042705 167777      BIC      #167777,R5     ;INTERESTED IN BIT 12
6316 037312 005705          TST      R5            ;BIT 12 SHOULD BE 0
6317 037314 001416          BEQ      8$            ;PASS
6318 037316 012737 001015 177746 MOV      #OFF,CCR       ;DISABLE CACHE
6319 037324 105037 177750      CLRB    CMR           ;CLEAR MAINT. MODE
6320 037330 010037 047334      MOV      R0,CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6321 037334 006237 047334      ASR     CA121
6322 037340 104406          ERROR          ;ERROR
        037342 037340          .WORD     .-2          ;-----
6323                                     ;VALID STORE MARCH PATTERN TEST- SET B
6324                                     ;READING CACHE VALID STORE DATA
6325                                     ;THRU CMR<12> DID NOT READ 0.
6326                                     ;THIS SUGGESTS THAT A RAM LOCATION
6327                                     ;SPECIFIED BY CA121 WAS OVERWRITTEN
6328                                     ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
6329                                     ;LOCATION.
6330                                     ;THIS INDICATES THAT VALID STORE RAM
6331                                     ;SET B IS BAD.
6332                                     ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6333 037344 047334          CA121
6334 037346 000000          .WORD     0
6335 037350 000430          BR       3$
6336 037352 042703 167777      8$:   BIC      #167777,R3     ;END THE TEST
6337 037356 022703 010000      CMP      #10000,R3     ;INTERESTED IN BIT 12 ONLY
6338 037362 001416          BEQ      9$            ;BIT 12 SHOULD BE 1
6339 037364 012737 001015 177746 MOV      #OFF,CCR       ;PASS
6340 037372 105037 177750      CLRB    CMR           ;DISABLE CACHE
6341 037376 010037 047334      MOV      R0,CA121      ;CLEAR MAINT. MODE
6342 037402 006237 047334      ASR     CA121          ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6343 037406 104406          ERROR          ;ERROR
        037410 037406          .WORD     .-2          ;-----
6344                                     ;VALID STORE MARCH PATTERN TEST- SET B
6345                                     ;READING CACHE VALID STORE DATA
6346                                     ;THRU CMR<12> DID NOT READ 1.
6347                                     ;THIS SUGGESTS THAT VALID STORE RAM
6348                                     ;IC SET B IS BAD.
6349                                     ;THRU CMR<12>
6350 037412 047334          CA121          ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6351 037414 000000          .WORD     0
6352 037416 000405          BR       3$
6353 037420 062700 000002      9$:   ADD      #2,R0
    
```

6354	037424	022700	100000			CMP	#100000,RO	.HAS ALL HI CACHE BEEN DONE?
6355	037430	001316				BNE	7\$	:NO,CONTINUE
6356	037432	012737	001015	177746	3\$:	MOV	#OFF,CCR	:DISABLE CACHE
6357	037440	105037	177750			CLRB	CMR	:DISABLE MAINT. MODE
6358	037444	000240			10\$:	NOP		:END OF TEST
	037446	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER



```

.SBTTL TEST # 204 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
*TEST 204 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*VERIFY THAT THE CACHE HIT NAND GATE CAN INDICATE A READ
*HIT CONDITION.
*PROCEDURE: CREATE A READ HIT CONDITION TO LO CACHE
*WITH LO CACHE ENABLED, AND VERIFY THAT
*OUTPUT OF THE CACHE HIT NAND GATE READS 0
*THRU CMR<8>.
*CONDITIONS: INPUTS CACHE HIT NAND GATE:
*COMPARE 1 =1
*COMPARE 2 =1
*COMPARE 3 =1
*VALID =1
*TAG PAR. ERR =1
*HI BYTE PE =1
*LO BYTE PE =1
*MISS HI =1
*MISS LO =1
*BYPASS/WRITE =1
*FAULT =1
*****

```

```

6380 037452 000004      TST204:      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
037454 037464          .WORD 40$          ;LOOP ON ERROR START LOCATION
037456 070004          .WORD 1$-40$+67764 ;SCOPE SYNC. LOCATION
037460 070030          .WORD 20$-40$+67764 ;LOOP ON ERROR END LOCATION
037462 070040          .WORD 25$-40$+67764 ;DISABLE CACHE
037464 012737 001015 177746 40$: MOV #OFF,CCR      ;LOCATE TEST CODE TO HIGH CACHE SPACE
037472 004437 002370          JSR R4,RELCTH     ;ADDRESS OF START OF NEXT TEST
037476 037570          .WORD 10$+2

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

6381 037500 005037 060000      CLR 60000          ;ALL 0'S TO MAIN MEMORY LOC. 60000
6382 037504 112737 000002 177750 1$: MOVB #HODO,CMR    ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6383                                ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6384                                ;GATE INTO CMR ONLY DURING THE DESTINATION
6385                                ;ACCESS OF AN INSTRUCTION.
6386 037512 012737 000011 177746      MOV #11,CCR        ;NO UCB SO AS TO WRITE CACHE STORES
6387                                ;ENABLE LOW CACHE FOR A READ HIT
6388 037520 005737 040000          TST 40000          ;
6389 037524 005737 060000          TST 60000          ;READ UPDATE TO LOW CACHE LOACATION 0000
6390 037530 005737 060000          20$: TST 60000     ;READ HIT; ALL INPUTS OF CACHE HIT NAND
6391                                ;GATE ARE 1; CLOCK STATUS OF NAND GATE
6392                                ;OUTPUT TO CMR<8>
6393 037534 013701 177750          MOV CMR,R1         ;SAVE CMR CONTENTS
6394 037540 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
037542 000240          NOP              ;FOR LOOP ON ERROR
6395 037544 012737 001015 177746      MOV #OFF,CCR        ;DISABLE CACHE
6396 037552 032701 000400          BIT #HIT,R1        ;WAS CACHE HIT SIGNAL A 0
6397 037556 001403          BEQ 10$           ;PASS
6398 037560 104406          ERROR           ;ERROR
                                ;-----

037562 037560          .WORD .-2

```

TEST # 204 - VERIFY CACHE VALID STORE RAM MEMORY IC'S

6399  
6400  
6401  
6402 037564 000000  
6403 037566 000240  
037570 005237 001472

10\$: .WORD 0  
NOP  
INC \$TESTN

:CACHE HIT TESTS  
:READING OUTPUT OF CACHE HIT NAND GATE  
:THRU CMR<B> DID NOT RESULT IN A 0  
:END OF TEST  
:INCREMENT TEST COUNTER

6423

```
.SBTTL TEST # 205 - CHECK FORCE MISS LOGIC.
*****
*TEST 205 - CHECK FORCE MISS LOGIC.
*CHECK FORCE MISS LOGIC.
*PROCEDURE: CREATE A READ HIT CONDITION TO HIGH CACHE
*           WITH 'FORCE MISS HI' DISABLED AND 'FORCE MISS
*           LO' ENABLED. VERIFY OUTPUT OF CACHE HIT NAND
*           GATE READS A 0 THRU CMR<8>.
*CONDITIONS: INPUTS CACHE HIT NAND GATE:
*           COMPARE 1 =1
*           COMPARE 2 =1
*           COMPARE 3 =1
*           VALID =1
*           TAG PAR. ERR =1
*           HI BYTE PE =1
*           LO BYTE PE =1
*           MISS HI =1
*           MISS LO =1
*           BYPASS/WRITE =1
*           FAULT =1
*****
```

```
6424 037574 000004 TST205: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
037576 037606 .WORD 40$ ;TEST START LOCATION
037600 060004 .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
037602 060030 .WORD 20$-40$+57764 ;SCOPE SYNC LOCATION
037604 060040 .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
037606 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
037614 004437 002342 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
037620 037712 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO LOW CACHE SPACE

6425 037622 005037 070000 CLR 7J000 ;ALL 0'S TO MAIN MEMORY LOC. 70000
6426 037626 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6427 ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6428 ;GATE INTO CMR ONLY DURING THE DESTINATION
6429 ;ACCESS OF AN INSTRUCTION.
6430 037634 012737 000005 177746 MOV #5,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6431 ;ENABLE HI CACHE FOR A READ HIT
6432 ;DISABLE LO CACHE
6433 037642 005737 050000 TST 50000 ;
6434 037646 005737 070000 TST 70000 ;READ UPDATE TO HI CACHE LOACATION 4000
6435 037652 005737 070000 20$: TST 70000 ;READ HIT; ALL INPUTS OF CACHE HIT NAND
6436 ;GATE ARE 1; CLOCK STATUS OF NAND GATE
6437 ;OUTPUT TO CMR<8>
6438 037656 013701 177750 MOV CMR,R1 ;SAVE CMR CONTENTS
6439 037662 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
037664 000240 NOP ;FOR LOOP ON ERROR
6440 037666 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
6441 037674 032701 000400 BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 0
6442 037700 001403 BEQ 10$ ;PASS
```

6443 037702 104406

ERROR

;ERROR  
;-----

037704 037702

.WORD    .-2

6444  
6445  
6446

;CACHE HIT TESTS  
;READING OUTPUT OF CACHE HIT NAND GATE  
;THRU CMR<8> DID NOT RESULT IN A 0

6447 037706 000000

.WORD    0

6448 037710 000240

10\$:

NOP

;END OF TEST

037712 005237 001472

INC

\$TESTN

;INCREMENT TEST COUNTER

.SBTTL TEST # 206 - CHK 'FORCE MISS LO' INHIBITS CACHE NAND GATE

\*\*\*\*\*  
 \*TEST 206 - CHK 'FORCE MISS LO' INHIBITS CACHE NAND GATE

VERIFY THAT 'FORCE MISS LO' WILL INHIBIT CACHE HIT NAND GATE  
 FROM INDICATING A CACHE HIT.

PROCEDURE: WITH 'FORCE MISS LO' ENABLED ATTEMPT A READ HIT TO LOW CACHE.  
 VERIFY THAT THE OUTPUT OF CACHE HIT NAND GATE  
 WILL READ AS A 1 THRU CMR<8>.

CONDITIONS: INPUTS CACHE HIT NAND GATE:

COMPARE 1 =1  
 COMPARE 2 =1  
 COMPARE 3 =1  
 VALID =1  
 TAG PAR. ERR =1  
 HI BYTE PE =1  
 LO BYTE PE =1  
 MISS HI =1  
 MISS LO =0  
 BYPASS/WRITE =1  
 FAULT =1

\*\*\*\*\*  
 TST206:

6469	037716	000004				SCPCND		;SCOPE CONDITIONS:GO SET UP FOR LOOP ON ;ERROR/LOOP ON TEST ;TEST START LOCATION ;LOOP ON ERROR START LOCATION ;SCOPE SYNC. LOCATION ;LOOP ON ERROR END LOCATION ;DISABLE CACHE ;LOCATE TEST CODE TO HIGH CACHE SPACE ;ADDRESS OF START OF NEXT TEST
	037720	037730				.WORD	40\$	
	037722	070004				.WORD	1\$-40\$+67764	
	037724	070030				.WORD	20\$-40\$+67764	
	037726	070040				.WORD	25\$-40\$+67764	
	037730	012737	001015	177746	40\$:	MOV	#OFF,CCR	
	037736	004437	002370			JSR	R4,RELCTH	
	037742	040034				.WORD	10\$+2	

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

6470	037744	005037	060000			CLR	60000	;ALL 0'S TO MAIN MEMORY LOC. 60000
6471	037750	112737	000002	177750	1\$:	MOV	#HODO,CMR	;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND ;CLOCKING OF OUTPUT OF CACHE HIT NAND ;GATE INTO CMR ONLY DURING THE DESTINATION ;ACCESS OF AN INSTRUCTION. ;NO UCB SO AS TO WRITE CACHE STORES ;DISABLE LOW CACHE
6472								
6473								
6474								
6475	037756	012737	000015	177746		MOV	#15,CCR	
6476								
6477	037764	005737	040000			TST	40000	
6478	037770	005737	060000			TST	60000	;READ UPDATE TO LOW CACHE LOACATION 0000
6479	037774	005737	060000		20\$:	TST	60000	;READ HIT; ;CLOCK STATUS OF NAND GATE ;OUTPUT TO CMR<8> ;SAVE CMR CONTENTS ;INSTRUCTION 'JMP 1\$' PLACED HERE ;FOR LOOP ON ERROR
6480								
6481								
6482	040000	013701	177750			MOV	CMR,R1	
6483	040004	000240			25\$:	NOP		
	040006	000240				NOP		
6484	040010	012737	001015	177746		MOV	#OFF,CCR	
6485	040016	032701	000400			BIT	#HIT,R1	;DISABLE CACHE ;WAS CACHE HIT SIGNAL A 1
6486	040022	001003				BNE	10\$	;PASS
6487	040024	104406				ERROR		;ERROR ;-----
6488	040026	040024				.WORD	.-2	;CACHE HIT TESTS

CKKKABO 11-44 KK11B CACHE MACRO M1113 27-OCT-80 08:39 PAGE 155-1 SEQUENCE 236  
TEST # 206 - CHK 'FORCE MISS LO' INHIBITS CACHE NAND GATE

6489  
6490  
6491 040030 000000  
6492 040032 000240  
040034 005237 001472

10\$: .WORD 0  
NOP  
INC \$TESTN

;READING OUTPUT OF CACHE HIT NAND GATE  
;THRU CMR<8> DID NOT RESULT IN A 1  
;END OF TEST  
;INCREMENT TEST COUNTER

```

.SBTTL TEST # 207 - CHK 'TAG PARITY ERROR' INHIBITS CACHE NAND GATE
*****
*TEST 207 - CHK 'TAG PARITY ERROR' INHIBITS CACHE NAND GATE
*   VERIFY THAT 'TAG PARITY ERROR' WILL INHIBIT CACHE HIT NAND GATE
*   FROM INDICATING A CACHE HIT.
*   CONDITIONS:      INPUTS CACHE HIT NAND GATE:
*                   COMPARE 1      =1
*                   COMPARE 2      =1
*                   COMPARE 3      =1
*                   VALID          =1
*                   TAG PAR. ERR    =0
*                   HI BYTE PE     =1
*                   LO BYTE PE     =1
*                   MISS HI        =1
*                   MISS LO        =1
*                   BYPASS/WRITE   =1
*                   FAULT          =1
*****
    
```

```

6510 040040 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. LOCATION
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
                                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                ;ARE RELOCATED TO HI CACHE SPACE
                                ;
040042 040052          .WORD 40$
040044 070000          .WORD 1$-40$+67764
040046 070054          .WORD 20$-40$+67764
040050 070072          .WORD 25$-40$+67764
040052 012737 001015 177746 40$: MOV #OFF,CCR
040060 004437 002370      JSR R4,RELCTH
040064 040220          .WORD 10$+2
                                ;O'S TO MAIN MEMORY LOCATION 0.
                                ;HODO ALLOWS UPDATES AND CACHE HITS
                                ; ONLY DURING THE DESTINATION ACCESS OF
                                ;AN INSTRUCTION.
                                ;NO UCB SO AS TO WRITE CACHE STORES
                                ;
                                ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
                                ;ALLOW WRITE WRONG PARITY TO TAGG PARITY STORE
                                ;READ UPDATE TO CACHE LOCATION 0000;
                                ;ALL 0'S(EVEN DATA) WILL BE WRITTEN TO DATA/TAG STORES.
                                ;0'S INTO DATA PARITY STORES,AND A 1
                                ;INTO TAG PARITY STORE.
                                ;DISABLE WWPT;ENABLE LOW CACHE
                                ;CLEAR CME
                                ;READ HIT; ALL 0'S WILL BE PLACED ON INPUTS
                                ;OF DATA PARITY ERROR CHECK PARITY GEN'S, BUT
                                ;TAG PARITY CHECK GENERATOR WILL
                                ;SEE ODD DATA DUE TO WRONG PARITY
                                ;FROM PREVIOUS READ UPDATE.
                                ;OUTPUT TO CMR<8>
                                ;SAVE CMR CONTENTS
                                ;DISABLE CACHE
                                ;INSTRUCTION 'JMP 1$' PLACED HERE
                                ;FOR LOOP ON ERROR
                                ;BEFORE LEAVING TEST ELIMINATE EFFECT OF WWPT
                                ;
6511 040066 005037 000000 1$: CLR 0
6512 040072 112737 000002 177750 MOVB #HODO,CMR
6513
6514
6515 040100 012737 000015 177746 MOV #15,CCR
6516 040106 005737 000000      TST 0
6517 040112 005737 040000      TST 40000
6518 040116 052737 002000 177746 BIS #WWPT,CCR
6519 040124 005737 000000      TST 0
6520
6521
6522
6523 040130 042737 002004 177746 BIC #WWPT+FMLO,CCR
6524 040136 005037 177744      CLR CMF
6525 040142 005737 000000 20$: TST 0
6526
6527
6528
6529
6530
6531 040146 013701 177750      MOV CMR,R1
6532 040152 012737 001015 177746 MOV #OFF,CCR
6533 040160 000240          NOP
                                25$: NOP
                                ;FOR LOOP ON ERROR
6534 040164 052737 000400 177746 BIS #FC,CCR
    
```

```

6535 040172 032737 010000 177746 500$: BIT #VCIP,CCR ;WAIT TILL DONE
6536 040200 001374 BNE 500$
6537 040202 032701 000400 BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
6538 040206 001003 BNE 10$ ;PASS
6539 040210 104406 ERROR ;ERROR
;-----
040212 040210 .WORD .-2
6540 ;CACHE HIT TESTS
6541 ;READING OUTPUT OF CACHE HIT NAND GATE
6542 ;THRU CMR<8> DID NOT RESULT IN A 1
6543 040214 000000 .WORD 0
6544 040216 000240 10$: NOP ;END OF TEST
040220 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
    
```



CKKKABO '11-44 KK11B CACHE  
TEST # 210 - CHK 'LO & HI BYTE PARITY ERROR'  
6561

MACRO M1113 27-OCT-80 08:39 PAGE 157 SEQUENCE 239

F 3

```
.SBTTL TEST # 210 - CHK 'LO & HI BYTE PARITY ERROR' STOPS NAND GATE
*****
*TEST 210 - CHK 'LO & HI BYTE PARITY ERROR' STOPS NAND GATE
*VERIFY THAT 'LO & HI BYTE PARITY ERROR' WILL INHIBIT CACHE HIT NAND GATE
*FROM INDICATING A CACHE HIT.
*CONDITIONS: INPUTS CACHE HIT NAND GATE:
*COMPARE 1 =1
*COMPARE 2 =1
*COMPARE 3 =1
*VALID =1
*TAG PAR. ERR =1
*HI BYTE PE =0
*LO BYTE PE =0
*MISS HI =1
*MISS LO =1
*BYPASS/WRITE =1
*FAULT =1
*****
```

```
TST210:
6562 040224 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
040226 040236          .WORD 40$          ;LOOP ON ERROR START LOCATION
040230 070000          .WORD 1$-40$+67764 ;SCOPE SYNC. LOCATION
040232 070054          .WORD 20$-40$+67764 ;LOOP ON ERROR END LOCATION
040234 070072          .WORD 25$-40$+67764 ;DISABLE CACHE
040236 012737 001015 177746 40$: MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE
040244 004437 002370     JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST
040250 040404          .WORD 10$+2

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

6563 040252 005037 000000 1$: CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
6564 040256 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS UPDATES AND CACHE HITS
6565                                     ; ONLY DURING THE DESTINATION ACCESS OF
6566                                     ;AN INSTRUCTION.
6567 040264 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6568 040272 005737 000000     TST 0 ;
6569 040276 005737 040000     TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
6570 040302 052737 000100 177746 BIS #WVPD,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO
6571                                     ; & HI BYTE PARITY STORE.
6572 040310 005737 000000     TST 0 ;READ UPDATE TO CACHE LOCATION 0000;
6573                                     ;ALL 0'S WILL BE WRITTEN TO DATA/TAG STORES,
6574                                     ;1'S INTO LO & HI BYTE DATA PARITY STORES,AND A 0
6575                                     ;INTO TAG PARITY STORE.
6576 040314 042737 000104 177746 BIC #WVPD+FMLO,CCR ;DISABLE WVPD;ENABLE LO CACHE
6577 040322 005037 177744     CLR CME ;CLEAR CME
6578 040326 005737 000000 20$: TST 0 ;READ HIT; ALL 0'S(EVEN DATA) WILL BI
6579                                     ;PLACED ON INPUTS
6580                                     ;OF TAG PARITY ERROR CHECK PARITY GEN'S, BUT
6581                                     ;LO & HI BYTE PARITY CHECK GENERATORS WILL
6582                                     ;SEE ODD DATA DUE TO WRONG PARITY
6583                                     ;FROM PREVIOUS READ UPDATE.
6584                                     ;CLOCK STATUS OF NAND GATE TO
6585                                     ;OUTPUT TO CMR<8>
6586 040332 013701 177750     MOV CMR,R1 ;SAVE CMR CONTENTS
6587 040336 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
```



H 3  
 .SBTTL TEST # 211 - 'FORCE MISS HI' INHIBITS NAND FROM IND. CACHE HIT  
 \*\*\*\*\*  
 \*TEST 211 - 'FORCE MISS HI' INHIBITS NAND FROM IND. CACHE HIT  
 \* VERIFY THAT 'FORCE MISS HI' WILL INHIBIT CACHE HIT NAND GATE  
 \* FROM INDICATING A CACHE HIT.  
 \* PROCEDURE: WITH 'FORCE MISS HI' ENABLED ATTEMPT A READ HIT TO HI CACHE.  
 \* VERIFY THAT THE OUTPUT OF CACHE HIT NAND GATE  
 \* WILL READ AS A 1 THRU CMR<8>.  
 \* CONDITIONS: INPUTS CACHE HIT NAND GATE:  
 \* COMPARE 1 =1  
 \* COMPARE 2 =1  
 \* COMPARE 3 =1  
 \* VALID =1  
 \* TAG PAR. ERR =1  
 \* HI BYTE PE =1  
 \* LO BYTE PE =1  
 \* MISS HI =0  
 \* MISS LO =1  
 \* BYPASS/WRITE =1  
 \* FAULT =1  
 \*\*\*\*\*

6620 040410 000004  
 040412 040422  
 040414 060004  
 040416 060030  
 040420 060040  
 040422 012737 001015 177746  
 040430 004437 002342  
 040434 040526

TST211:  
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
 ;ERROR/LOOP ON TEST  
 .WORD 40\$ ;TEST START LOCATION  
 .WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION  
 .WORD 20\$-40\$+57764 ;SCOPE SYNC LOCATION  
 .WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION  
 MOV #OFF,CCR ;DISABLE CACHE  
 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE  
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO LOW CACHE SPACE

6621 040436 005037 070000  
 6622 040442 112737 000002 177750  
 6623  
 6624  
 6625  
 6626 040450 012737 000015 177746  
 6627  
 6628 040456 005737 050000  
 6629 040462 005737 070000  
 6630 040466 005737 070000  
 6631  
 6632 040472 013701 177750  
 6633 040476 000240  
 040500 000240  
 6634 040502 012737 001015 177746  
 6635 040510 032701 000400  
 6636 040514 001003  
 6637 040516 104406

CLR 70000 ;ALL 0'S TO MAIN MEMORY LOC. 70000  
 MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND  
 ;CLOCKING OF OUTPUT OF CACHE HIT NAND  
 ;GATE INTO CMR ONLY DURING THE DESTINATION  
 ;ACCESS OF AN INSTRUCTION.  
 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES  
 ;DISABLE HI CACHE  
 TST 50000  
 TST 70000 ;READ UPDATE HI CACHE LOCATION 4000  
 TST 70000 ;READ HIT; CLOCK STATUS OF NAND GATE  
 ;OUTPUT TO CMR<8>  
 MOV CMR,R1 ;SAVE CMR CONTENTS  
 ;INSTRUCTION 'JMP 1\$' PLACED HERE  
 ;FOR LOOP ON ERROR  
 MOV #OFF,CCR ;DISABLE CACHE  
 BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1  
 BNE 10\$ ;PASS  
 ERROR ;ERROR  
 ;-----

040520 040516  
 6638  
 6639

.WORD .-2  
 ;CACHE HIT TESTS  
 ;READING OUTPUT OF CACHE HIT NAND GATE

6640  
6641 040522 000000  
6642 040524 000240  
040526 005237 001472

10\$:      .WORD 0  
          NOP  
          INC     \$TESTN

;THRU CMR<8> DID NOT RESULT IN A 1  
;END OF TEST  
;INCREMENT TEST COUNTER



6683  
6684  
6685 040652 000000  
6686 040654 000240  
          040656 005237 001472

108:      .WORD    0  
          NOP  
          INC     \$TESTN

:READING OUTPUT OF CACHE HIT NAND GATE  
:THRU CMR<8> DID NOT RESULT IN A 1  
  
:END OF TEST  
:INCREMENT TEST COUNTER

L 3

SBTTL TEST # 213 - 'VALID' INPUT TO NAND INHIBITS IND. CACHE HIT  
 \*\*\*\*\*  
 \*TEST 213 - 'VALID' INPUT TO NAND INHIBITS IND. CACHE HIT  
 \*VERIFY THAT 'VALID' INPUT TO CACHE HIT NAND GATE WILL INHIBIT NAND  
 \*GATE FROM INDICATING A CACHE HIT.  
 \*PROCEDURE: CREATE A CONDITION WHERE ONLY VALID INPUT ON  
 \*CACHE HIT NAND GATE INHIBITS NAND GATE:  
 \* 1.UPDATE CACHE LOCATION 0000  
 \* 2.CAUSE INVALIDATION BY A WRITE HIT  
 \* IN BYPASS MODE  
 \* 3.CAUSE READ HIT  
 \*VERIFY THAT OUTPUT OF NAND GATE  
 \*WILL READ AS A 1 THRU CMR<8>.  
 \*CONDITIONS: INPUTS CACHE HIT NAND GATE:  
 \* COMPARE 1 =1  
 \* COMPARE 2 =1  
 \* COMPARE 3 =1  
 \* VALID =0  
 \* TAG PAR. ERR =1  
 \* HI BYTE PE =1  
 \* LO BYTE PE =1  
 \* MISS HI =1  
 \* MISS LO =1  
 \* BYPASS/WRITE =1  
 \* FAULT =1  
 \*\*\*\*\*

```

TST213:
6712 040662 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          .WORD 40$          ;ERROR/LOOP ON TEST
          .WORD 1$-40$+67764 ;TEST START LOCATION
          .WORD 20$-40$+67764 ;LOOP ON ERROR START LOCATION
          .WORD 25$-40$+67764 ;SCOPE SYNC. LOCATION
          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
          .WORD 25$-40$+67764 ;DISABLE CACHE
040664 040674          .MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE
040666 070012          .JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST
040670 070054          .WORD 10$+2
040672 070064
040674 012737 001015 177746 40$:
040702 004437 002370
040706 041024

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

6713 040710 005000          CLR R0          ;CLEAR R0
6714 040712 005037 060000  CLR 60000      ;ALL 0'S TO MAIN MEMORY LOC. 60000
6715 040716 012701 060000  MOV #60000,R1  ;ADDRESS 60000 TO R1
6716 040722 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6717          ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6718          ;GATE INTO CMR ONLY DURING THE DESTINATION
6719          ;ACCESS OF AN INSTRUCTION.
6720 040730 012737 000011 177746  MOV #11,CCR   ;NO UCB SO AS TO WRITE CACHE STORES
6721          ;ENABLE LO CACHE
6722 040736 005737 040000  TST 40000
6723 040742 005737 060000  TST 60000      ;READ UPDATE; ASSURE CORRECT PARITY IS WRITTEN
6724          ;FOR CACHE LOCATION 0000
6725 040746 052737 001000 177746  BIS #UCB,CCR  ;SET UCB SO AS TO INVALIDATE CACHE LOCATIONS
6726          ;DURING WRITE HIT
6727 040754 010011          MOV RC,(R1)   ;CAUSE WRITE HIT TO LOCATION 60000;
6728          ;UCB CAUSES CACHE LOC. 0000 TO BE INVALIDATED
6729 040756 042737 001000 177746  BIC #UCB,CCR  ;CLEAR UCB
  
```









```

.SBTTL TEST # 215 - 'COMPARE 2' INPUT STOPS NAND FROM IND. HIT
*****
*TEST 215 - 'COMPARE 2' INPUT STOPS NAND FROM IND. HIT
*VERIFY THAT 'COMPARE 2' INPUT TO CACHE HIT NAND GATE CAN INHIBIT
* NAND GATE FROM INDICATING A CACHE HIT.
*PROCEDURE: CREATE A READ UPDATE TO LOW CACHE CAUSED BY ONLY
* BIT 14 ON CACHE ADDRESS LINE BEING DIFFERENT
* FROM BIT 14 IN TAG STORE. VERIFY THAT OUTPUT OF CACHE HIT
* NAND GATE WILL READ AS A 1 THRU CMR<8>.
*CONDITIONS: INPUTS CACHE HIT NAND GATE:
* COMPARE 1 =1
* COMPARE 2 =0
* COMPARE 3 =1
* VALID =1
* TAG PAR. ERR =1
* HI BYTE PE =1
* LO BYTE PE =1
* MISS HI =1
* MISS LO =1
* BYPASS/WRITE =1
* FAULT =1
*****

```

```

6830 041242 000004 TST215: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
041244 041254 .WORD 40$ ;TEST START LOCATION
041246 070000 .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
041250 070024 .WORD 20$-40$+67764 ;SCOPE SYNC. LOCATION
041252 070034 .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
041254 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
041262 004437 002370 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
041266 041354 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE

6831 041270 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS,UPDATES, AND
6832 ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6833 ;GATE INTO CMR ONLY DURING THE DESTINATION
6834 ;ACCESS OF AN INSTRUCTION.
6835 041276 012737 000011 177746 MOV #11,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6836 ;ENABLE LOW CACHE
6837 041304 005737 040000 TST 40000 ;
6838 041310 005737 000000 TST 0 ;READ UPDATE; ASSURE ALL 0'S IN TAG STORE
6839 ;LOCATION 0000,AND CORRECT PARITY IS WRITTEN.
6840 041314 005737 040000 20$: TST 40000 ;READ UPDATE TO CACHE LOCATION 0000
6841 ;CAUSED BY BIT 14 ON CACHE ADDRESS LINE
6842 ;DIFFERENT FROM TAG STORE BIT 14.
6843 ;CLOCK STATUS OF NAND GATE
6844 ;OUTPUT TO CMR<8>
6845 041320 013701 177750 MOV CMR,R1 ;SAVE CMR CONTENTS
6846 041324 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
041326 000240 NOP ;FOR LOOP ON ERROR
6847 041330 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
6848 041336 032701 000400 BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
6849 041342 001003 BNE 10$ ;PASS
6850 041344 104406 ERROR ;ERROR

```

6851 041346 041344  
6852  
6853  
6854 041350 000000  
6855 041352 000240  
          041354 005237 001472

.WORD   .-2  
  
10\$:    .NOP  
          INC    \$TESTN

;-----  
;CACHE HIT TESTS  
;READING OUTPUT OF CACHE HIT NAND GATE  
;THRU CMR<8> DID NOT RESULT IN A 1  
  
;END OF TEST  
;INCREMENT TEST COUNTER

```

.SBTTL TEST # 216 - 'COMPARE 3' INPUT STOPS GATE FROM IND. HIT
*****
*TEST 216 - 'COMPARE 3' INPUT STOPS GATE FROM IND. HIT
*   VERIFY THAT 'COMPARE 3' INPUT TO CACHE NAND GATE WILL INHIBIT NAND GATE
*   FROM INDICATING A CACH HIT.
*   PROCEDURE:   CREATE A READ UPDATE TO LO CACHE CAUSED BY
*                 ONLY BIT 13 ON CACHE ADDRESS LINE BEING
*                 DIFFERENT FROM BIT 13 IN TAG STORE.
*                 VERIFY THAT THE OUTPUT OF CACHE HIT NAND GATE
*                 WILL READ AS A 1 THRU CMR<8>.
*   CONDITIONS:  INPUTS CACHE HIT NAND GATE:
*                 COMPARE 1   =1
*                 COMPARE 2   =1
*                 COMPARE 3   =0
*                 VALID       =1
*                 TAG PAR. ERR =1
*                 HI BYTE PE  =1
*                 LO BYTE PE  =1
*                 MISS HI     =1
*                 MISS LO     =1
*                 BYPASS/WRITE =1
*                 FAULT       =1
*****

```

```

6878 041360 000004      TST216:      SCPCND          ;SCOPE CONDITIONS:GO SET JP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. LOCATION
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
                                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                ;ARE RELOCATED TO HI CACHE SPACE
041362 041372          .WORD 40$
041364 070000          .WORD 1$-40$+67764
041366 070024          .WORD 20$-40$+67764
041370 070034          .WORD 25$-40$+67764
041372 012737 001015 177746 40$:  MOV #OFF,CCR
041400 004437 002370      JSR R4,RELCTH
041404 041472          .WORD 10$+2
                                ;HODO ALLOWS READ HITS,UPDATES, AND
                                ;CLOCKING OF OUTPUT OF CACHE HIT NAND
                                ;GATE INTO CMR ONLY DURING THE DESTINATION
                                ;ACCESS OF AN INSTRUCTION.
                                ;NO UCB SO AS TO UPDATE CACHE STORES
                                ;ENABLE LOW CACHE
                                ;READ UPDATE; ASSURE ALL 0'S IN TAG
                                ;STORE LOCATION 0000 AND CORRECT PARITY
                                ;IS WRITTEN.
                                ;READ UPDATE TO CACHE LOC. 0000 CAUSED BY
                                ;BIT 13 ON CACHE ADDRESS LINES BEING
                                ;DIFFERENT FROM BIT 13 IN TAG STORE.
                                ;CLOCK STATUS OF CACHE HIT NAND GATE
                                ;OUTPUT TO CMR<8>
                                ;SAVE CMR CONTENTS
                                ;INSTRUCTION 'JMP 1$' PLACED HERE
                                ;FOR LOOP ON ERROR
                                ;DISABLE CACHE
                                ;WAS CACHE HIT SIGNAL A 1
6879 041406 112737 000002 177750 1$:  MOVB #HODO,CMR
6880
6881
6882
6883 041414 012737 000011 177746      MCV #11,CCR
6884
6885 041422 005737 020000          TST 20000
6886 041426 005737 000000          TST 0
6887
6888
6889 041432 005737 020000          20$: TST 20000
6890
6891
6892
6893
6894 041436 013701 177750          MOV CMR,R1
6895 041442 000240          25$: NOP
                                NOP
6896 041446 012737 001015 177746      MOV #OFF,CCR
6897 041454 032701 000400          BIT #HIT,R1

```

6898	041460	001003		BNE	10\$			:PASS
6899	041462	104406		ERROR				:ERROR
	041464	041462		.WORD	.-2			:-----
6900								:CACHE HIT TESTS
6901								:READING OUTPUT OF CACHE HIT NAND GATE
6902								:THRU CMR<8> DID NOT RESULT IN A 1
6903	041466	000000		.WORD	0			
6904	041470	000240		NOP				:END OF TEST
	041472	005237	001472	10\$:	INC	\$TESTN		:INCREMENT TEST COUNTER

SBTTL TEST # 217 - CACHE READ HIT RESULTS IN PROPER OUTPUT

\*\*\*\*\*

\*TEST 217 - CACHE READ HIT RESULTS IN PROPER OUTPUT

\* VERIFY THAT A CACHE READ HIT WILL RESULT IN DATA BEING READ  
\* FROM CACHE DATA STORE, ASSURING THAT THE CACHE HAS ISSUED A  
\* A CPU CLOCK RESTART SIGNAL. ASSURE THAT ALL 0'S CAN BE CACHED  
\* OUT OF CACHE DATA STORE.

\*\*\*\*\*

TST217:

6912	041476	000004			SCPCND		;SCOPE CONDITIONS:GO SET UP FOR LOOP ON ;ERROR/LOOP ON TEST	
	041500	041510			.WORD	40\$	;TEST START LOCATION	
	041502	070000			.WORD	1\$-40\$+67764	;LOOP ON ERROR START LOCATION	
	041504	000000			.WORD	0	;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST	
	041506	070106			.WORD	25\$-40\$+67764	;LOOP ON ERROR END LOCATION	
	041510	012737	001015	177746	40\$:	MOV	#OFF,CCR	;DISABLE CACHE
	041516	004437	002370			JSR	R4,RELCTH	;LOCATE TEST CODE TO HIGH CACHE SPACE
	041522	041754			.WORD	10\$+2	;ADDRESS OF START OF NEXT TEST	

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

6913	041524	012701	040000		1\$:	MOV	#40000,R1	;ADDRESS 40000 TO R1
6914	041530	012702	060000			MOV	#60000,R2	;ADDRESS 60000 TO R2
6915	041534	012737	070076	000014		MOV	#3\$-1\$+70000,14	;SETUP BPT TRAP VECTORS
6916	041542	012737	000340	000016		MOV	#340,16	
6917	041550	005037	002070			CLR	FAIL1	;CLEAR ERROR FLAGS
6918	041554	005037	002072			CLR	FAIL2	
6919	041560	012706	060002			MOV	#60002,SP	;STACK POINTER NOW POINTS TO ADDRESS 60002
6920	041564	005037	060000			CLR	60000	;PRECONDITION MAIN MEMORY ADDRESS LOCATION ;60000 WITH ALL 0'S
6921								
6922	041570	012737	000340	177776		MOV	#340,PSW	;PRECONDITION PSW TO 340
6923	041576	112737	000002	177750		MOVB	#HODO,CMR	;HODO WILL ALLOW READ HITS AND UPDATES ;ONLY DURING THE DESTINATION MEMORY ACCESS ;OF AN INSTRUCTION.
6924								
6925								
6926								
6927								
6928	041604	012737	000011	177746		MOV	#11,CCR	;HODO DOES NOT ALLOW A CACHE UPDATE ;TO OCCUR DUE TO WRITE UPDATES. ;NO BYPASS TO ALLOW WRITES TO CACHE STORES.
6929								;ENABLE LOW CACHE
6930	041612	000257				CCC		;CLEAR ALL CONDITION CODES
6931	041614	005711				TST	(R1)	
6932	041616	005712				TST	(R2)	
6933								;CACHE READ UPDATE. WRITE ALL 0'S FROM ;MAIN MEMORY LOCATION TO CACHE DATA STORE ;LOCATION 0000.
6934								
6935	041620	000003				BPT		;BREAKPOINT TRAP. DUE TO A TRAP,THE PSW ;WILL BE WRITTEN TO THE STACK, WHICH NOW ;POINTS TO ADDRESS 60000.THE TRAP INSTRUCTION ;IS A NON-DESTINATION ACCESS INSTR.
6936								;SINCE HODO IS BEING USED, A CACHE UPDATE ;WILL BE INHIBITED. MAIN MEMORY ;ADDRESS 60000 WILL CONTAIN PSW DATA OF 344,AND ;THE LOCATION IN CACHE CORRESPONDING TO ADDRESS ;60000 WILL BE LEFT WITH ALL 0'S DATA.
6937								
6938								
6939								
6940								
6941								
6942								
6943								
6944	041622	042737	000002	177750	3\$:	BIC	#HOD',CMR	;TRAP TO HERE;DISABLE HODO
6945	041630	011200				MOV	(R2 ,R0	; WHEN THIS INSTRUCTION READS ;ADDRESS 60000
6946								
6947								;A CACHE READ HIT SHOULD RESULT AND A CPU CLOCK





6988

```
.SBTTL TEST # 220 - FLOATING 1 CAN BE CACHED FROM DATA STORE
:*****
:*TEST 220 - FLOATING 1 CAN BE CACHED FROM DATA STORE
:*  VERIFY THAT A FLOATING 1 ACROSS 0'S DATA PATTERN CAN BE CACHED
:*  FROM CACHE DATA STORE.
:*****
```

6989 041760 000004

041760 041772  
 041764 070016  
 041766 000000  
 041770 070040  
 041772 012737  
 042000 004437  
 042004 042122

001015 177746 40\$:  
 002370

```
TST220:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
;WORD 10$+2
```

```
;THE FOLLOWING LOCATIONS INCLUDING 10$
;ARE RELOCATED TO HI CACHE SPACE
```

6990 042006 012701 040000  
 6991 042012 012702 060000  
 6992 042016 012737 000001 047344  
 6993 042024 013737 047344 060000 1\$:  
 6994  
 6995 042032 012737 000011 177746  
 6996  
 6997 042040 005711  
 6998 042042 005712  
 6999  
 7000  
 7001 042044 011200  
 7002  
 7003  
 7004  
 7005  
 7006 042046 000240  
 042050 000240  
 7007 042052 012737 001015 177746  
 7008 042060 020037 047344  
 7009 042064 001412  
 7010 042066 010037 047346  
 7011 042072 013737 047344 047332  
 7012 042100 104406

```
MOV #40000,R1 ;ADDRESS 40000 TO R1
MOV #60000,R2 ;ADDRESS 60000 TO R2
MOV #1,FLTPAT ;1ST FLOATING 1 PATTERN: 000001
MOV FLTPAT,60000 ;WRITE FLOATING 1 PATTERN TO MAIN MEMORY
;LOCATION 60000
MOV #11,CCR ;NO BYPASS TO ALLOW WRITES TO CACHE STORES.
;ENABLE LOW CACHE
TST (R1)
TST (R2) ;CACHE READ UPDATE. WRITE FLOATING 1 PATTERN FROM
;MAIN MEMORY LOCATION TO CACHE DATA STORE
;LOCATION 0000.
MOV (R2),R0 ;WHEN THIS INSTRUCTION READS
;ADDRESS 60000,A CACHE READ HIT SHOULD RESULT
;AND A CPU CLOCK RESTART SIGNAL SHOULD BE ISSUED.
;THE CPU SHOULD READ FLOATING 1 PATTERN FROM CACHE DATA STOR
;RATHER THAN MAIN MEMORY.
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
MOV #OFF,CCR ;DISABLE CACHE
CMP R0,FLTPAT ;WAS THE CORRECT FLOATING 1 PATTERN RECEIVED
BEQ 9$ ;PASS
MOV R0,RECDAT ;GET DATA RECEIVED
MOV FLTPAT,EXDAT6 ;GET EXPECTED DATA
ERROR ;ERROR
;-----
```

042102 042100  
 7013  
 7014  
 7015  
 7016  
 7017 042104 047332  
 7018  
 7019 042106 047346  
 7020 042110 000000

```
.WORD -2
EXDAT6
RECDAT
;CPU CLOCK RESTART-CACHED DATA TESTS
;CREATING A READ HIT BY READING ADDRESS 60000
;RESULTED IN INCORRECT FLOATING 1 PATTERN
;BEING CACHED FROM LACHE DATA STORE
;PRINT FLOATING 1 PATTERN EXPECTED FROM THE
;READ HIT TO ADDRESS 60000
;PRINT DATA RECEIVED FROM READ HIT TO ADDRESS 60000
;WORD 0
```

7021	042112	006337	047344	9\$:	ASL	FLTPAT	:NEXT FLOATING 1 PATTERN
7022	042116	103342			BCC	1\$	:IF FLOATING 1 PATTERN 10000 HAS NOT BEEN
7023							:DONE ,CONTINUE.
7024	042120	000240		10\$:	NOP		:END OF TEST
	042122	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

TEST # 221 - DMA WRITE HITS STOPS ALL CACHE STORE LOCS  
7029

.SBTTL TEST # 221 - DMA WRITE HITS STOPS ALL CACHE STORE LOCS  
:\*\*\*\*\*  
:TEST 221 - DMA WRITE HITS STOPS ALL CACHE STORE LOCS  
:\* VERIFY THAT DMA WRITE HITS WILL INVALIDATE ALL OF LOW CACHE VALID  
:\* STORE LOCATIONS.  
:\*\*\*\*\*

```

7030 042126 000004          SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          .WORD 40$          ;ERROR/LOOP ON TEST
          .WORD 1$-40$+67764 ;TEST START LOCATION
          .WORD 0           ;LOOP ON ERROR START LOCATION
          .WORD 25$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
          MOV #OFF,CCR      ;DISABLE CACHE
          JSR R4,RELCTH    ;LOCATE TEST CODE TO HIGH CACHE SPACE
          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST
    
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

```

7031 042154 032737 020000 177746          BIT #VSIU,CCR          ;IS SET A BEING USED?
7032 042162 001407          BEQ 7$                ;YES
7033 042164 052737 000400 177746          BIS #FC,CCR           ;NO; FLUSH CACHE FOR SET A
7034 042172 032737 010000 177746 500$:  BIT #VCIP,CCR         ;WAIT TILL FLUSH COMPLETE
7035 042200 001374          BNE 500$              ;
7036 042202 012705 060000          7$:  MOV #60000,R5          ;;ADDRESS 60000 INTO R5
7037 042206 012703 040000          MOV #40000,R3         ;ADDRESS 40000 INTO R3
7038 042212 132737 000200 001507          BITB #APTSIZE,$ENVM  ;WILL APT SIZE?
7039 042220 001405          BEQ 3$                ;NO,GO AUTOSIZE
7040 042222 032737 010000 001512          BIT #10000,$USWR     ;DOES APT SAY TO PERFORM TEST
7041 042230 001426          BEQ 11$               ;APT SAYS DO NOT PERFORM TEST
7042 042232 000504          BR 1$                ;APT SAYS DO TEST
7043 042234 012737 070116 000004 3$:  MOV #5$-40$+67764,4 ;SETUP FOR TRAP
7044 042242 012737 000340 000006          MOV #340,6
7045 042250 005737 170006          TST BECR1           ;ACCESS UNIBUS EXERCISER
7046 042254 000240          NOP
7047 042256 012737 000006 000004          MOV #6,4            ;RESTORE VECTORS
7048 042264 005037 000006          CLR 6
7049 042270 000465          BR 1$                ;UNIBUS EXERCISER IS PRESENT;PROCEED WITH TEST
7050 042272 022626          5$:  CMP (SP)+,(SP)+     ;TRAP RETURN;EXERCISER NOT PRESENT
7051 042274 012737 000006 000004          MOV #6,4            ;RESTORE VECTORS
7052 042302 005037 000006          CLR 6
7053 042306 005737 001474          11$: TST $PASS           ;IS THIS SFIRST PASS?
7054 042312 001156          BNE 10$              ;SKIP MESSAGE;SKIP TEST
7055 042314 023737 000042 000046          CMP 42,46          ;IS THIS ACT11 QV OR AUTO ACCEPT?
7056 042322 001552          BEQ 10$              ;YES SKIP TYPEOUT
7057 042324 104401 042332          TYPE ,65$          ;;TYPE ASCIZ STRING
          042330 000402          BR 64$              ;;GET OVER THE ASCIZ
          042332 200 200 000 65$: .ASCIZ <CRLF><CRLF>
          .EVEN
          64$:
7058 042336          TYPE ,67$          ;;TYPE ASCIZ STRING
          042342 000432          BR 66$              ;;GET OVER THE ASCIZ
          042344 125 116 111 67$: .ASCIZ /UNIBUS EXERCISER NOT USED- DMA TESTS NOT PERFORMED/
          .EVEN
          66$:
7059 042430          TYPE ,69$          ;;TYPE ASCIZ STRING
          104401 042436
    
```

```

TEST # 221 - DMA WRITE HITS STOPS ALL CACHE STORE LOCS
042434 000402 BR 68$ ;:GET OVER THE ASCIIZ
042436 200 200 000 69$: .ASCIIZ <CRLF><CRLF>
.EVEN
042442 68$:
7060 042442 000502 BR 10$
7061 042444 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
7062 042452 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS UPDATES ONLY DURING THE
7063 ;DESTINATION ACCESS OF AN INSTRUCTION
7064 042460 005723 2$: TST (R3)+
7065 042462 005725 TST (R5)+ ;UPDATE ALL LOW CACHE LOCATIONS MAKING
7066 ;ALL VALID STORE LOCATIONS =1
7067 042464 022705 070000 CMP #70000,R5 ;COMPLETE?
7068 042470 001373 BNE 2$ ;NO
7069 042472 042737 000002 177750 BIC #HODO,CMR ;CLEAR HODO SO VALID STORE CAN BE WRITTEN
7070 ;BY UNIBUS EXERCISER.
7071 042500 012705 060000 MOV #60000,R5 ;ADDRESS 60000 INTO R5
7072 042504 012737 060000 170004 MOV #60000,BEBA ;SETUP UNIBUS EXERCISER
7073 ;ADRESS
7074 042512 012737 174000 170002 MOV #-4000,BECC ;TRANSFER COUNT
7075 042520 012737 177777 170000 MOV #177777,BEDA ;DATA FOR WRITE XFER
7076 042526 012737 000000 170016 MOV #0,BECC2 ;SETUP CONTROL REGISTER 2
7077 042534 012737 003045 170006 MOV #3045,BECC1 ;SETUP CONTROL REGISTER 1;START XFER
7078 042542 105737 170006 4$: TSTB BECC1 ;WAIT FOR EXERCISER TO COMPLETE
7079 042546 100375 BPL 4$
7080 042550 052737 000002 177750 6$: BIS #HODO,CMR ;IMPLEMENT HODO. ALLOWS VALID STORE
7081 ;DATA TO BE WRITTEN TO CMR<12> ONLY
7082 ;DURING THE DESTINATION MEMORY ACCESS
7083 ;OF AN INSTRUCTION.
7084 042556 005715 TST (R5) ;READ LOW CACHE ADDRESS
7085 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
7086 ;WRITE VALID STORE DATA INTO CMR<12>
7087 ;FROM VALID STORE ADDRESS LOCATION
7088 ;JUST READ.
7089 042560 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
7090 042564 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
042566 000240 NOP ;FOR LOOP ON ERROR
7091 042570 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
7092 042576 105037 177750 CLRB CMR ;DISABLE MAINTENANCE
7093 042602 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 0.
7094 042606 001411 BEQ 9$ ;PASS
7095 042610 010537 047334 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
7096 ;USED: CA<12:1>
7097 042614 006237 047334 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
7098 042620 104406 ERROR ;ERROR
;-----
042622 042620 .WORD -2
7099 ;DMA TESTS
7100 ;READING VALID STORE DATA
7101 ;THRU CMR<12> DID NOT RESULT IN 0.
7102 ;THIS INDICATES THAT VALID STORE WAS
7103 ;NOT INVALIDATED DUE TO DMA WRITE HIT.
7104 042624 047334 CA121 ;PRINT VALID STORE ADDRESS LOCATION
7105 ;USED: CA<12:1>.
7106 042626 000000 .WORD 0
7107 042630 000407 BR 10$ ;IF ERROR END TEST
7108 042632 062705 000002 9$: ADD #2,R5 ;NEXT VALID STORE LOCATION
7109 042636 062703 000002 ADD #2,R3

```

7110	042642	020527	070000		CMP	R5,#70000	;HAVE ALL LOW CACHE ADDRESS LOCATIONS
7111							;BEEN DONE?
7112	042646	001340			BNE	6\$	;NO
7113	042650	000240		10\$:	NOP		;END OF TEST
	042652	005237	001472		INC	\$TESTN	;INCREMENT TEST COUNTER

N 4

\*\*\*\*\*  
 .SBTTL TEST # 222 - DMA WRITE CAUSES TIMEOUT & CCR REG NOT ALTERED  
 \*\*\*\*\*  
 \*TEST 222 - DMA WRITE CAUSES TIMEOUT & CCR REG NOT ALTERED  
 \* VERIFY THAT A DMA WRITE TO CCR REGISTER WILL RESULT IN A TIMEOUT  
 \* AND THAT THE CCR REGISTER WILL NOT BE ALTERED  
 \*\*\*\*\*  
 TST222:

```

7119 042656 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                ;DISABLE CACHE
                                ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
                                40$
                                .WORD 40$
                                .WORD 1$-40$+67764
                                .WORD 0
                                .WORD 25$-40$+67764
                                MOV #OFF,CCR
                                JSR R4,RELCTH
                                .WORD 10$+2
    
```

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
 ;ARE RELOCATED TO HI CACHE SPACE

```

7120 042704 032737 020000 177746          BIT #VSIU,CCR          ;IS SET A BEING USED?
7121 042712 001407          BEQ 7$                ;YES
7122 042714 052737 000400 177746          BIS #FC,CCR           ;NO; FLUSH CACHE FOR SET A
7123 042722 032737 010000 177746 500$: BIT #VCIP,CCR         ;WAIT TILL FLUSH COMPLETE
7124 042730 001374          BNE 500$
7125 042732 012705 060000          7$: MOV #60000,R5        ;:ADDRESS 60000 INTO R5
7126 042736 012703 040000          MOV #40000,R3        ;:ADDRESS 40000 INTO R3
7127 042742 132737 000200 001507          BITB #APTSIZE,$ENVM  ;:WILL APT SIZE?
7128 042750 001405          BEQ 3$                ;:NO,GO AUTOSIZE
7129 042752 032737 010000 001512          BIT #10000,$USWR     ;:DOES APT SAY TO PEPFORM TEST
7130 042760 001426          BEQ 11$              ;:APT SAYS DO NOT PERFORM TEST
7131 042762 000426          BR 1$                ;:APT SAYS DO TEST
7132 042764 012737 070116 000004 3$: MOV #5$-40$+67164,4 ;:SETUP FOR TRAP
7133 042772 012737 000340 000006          MOV #340,6
7134 043000 005737 170006          TST BECR1           ;:ACCESS UNIBUS EXERCISER
7135 043004 000240          NOP
7136 043006 012737 000006 000004          MOV #6,4            ;:RESTORE VECTORS
7137 043014 005037 000006          CLR 6
7138 043020 000407          BR 1$                ;:UNIBUS EXERCISER IS PRESENT;PROCEED WITH TEST
7139 043022 022626          5$: CMP (SP)+,(SP)+  ;:TRAP RETURN;EXERCISER NOT PRESENT
7140 043024 012737 000006 000004          MOV #6,4            ;:RESTORE VECTORS
7141 043032 005037 000006          CLR 6
7142 043036 000502          11$: BR 10$           ;:SKIP TEST
7143 043040 012737 000015 177746 1$: MOV #15,CCR         ;:CACHE OFF-DISABLE INTERRUPT
7144 043046 012737 070256 000510          MOV #6$-40$+67764,510 ;:SETUP RETURN ADDRESS FOR
7145          ;:A UNIBUS EXER, TRAP
7146 043054 012737 000340 000512          MOV #340,512
7147 043062 012737 177746 170004          MOV #177746,BEBA    ;:SETUP UNIBUS EXERCISER ADRSS
7148 043070 012737 177777 170002          MOV #-1,BECC        ;:TRANSFER COUNT
7149 043076 012737 001015 170000          MOV #1015,BEDA      ;:DATA FOR WRITE XFER
7150 043104 012737 000003 170016          MOV #3,BECC2        ;:SETUP CONTROL REGISTER 2
7151 043112 012737 003045 170006          MOV #3045,BECC1     ;:SETUP CONTROL REGISTER 1;START XFER
7152 043120 105737 170006          4$: TSTB BECR1      ;:WAIT FOR EXERCISER TO COMPLETE
7153 043124 100375          BPL 4$
7154 043126 012737 001000 002062          MOV #1000,LOOP      ;:GIVE ENOUGH TIME FOR TIMEOUT TO OCCUR
7155 043134 005337 002062          2$: DEC LOOP
7156 043140 001375          BNE 2$
    
```



7189

.SBTTL TEST # 223 - ALL 6 HIT REG BITS READ 0 DUE TO 6 READ MISSES

\*\*\*\*\*  
\*TEST 223 - ALL 6 HIT REG BITS READ 0 DUE TO 6 READ MISSES  
\* CHECK THAT ALL SIX HIT REGISTER BITS CAN READ 0 DUE TO SIX  
\* READ MISSES  
\*\*\*\*\*

043252  
7190 043252 000004

043254 043264  
043256 070010  
043260 000000  
043262 070046  
043264 012737  
043272 004437  
043276 043412

001015 177746 40\$:  
002370

SCPCND  
.WORD 40\$  
.WORD 1\$-40\$+67764  
.WORD 0  
.WORD 25\$-40\$+67764  
MOV #OFF,CCR  
JSR R4,RELCTH  
.WORD 10\$+2

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
:ERROR/LOOP ON TEST  
:TEST START LOCATION  
:LOOP ON ERROR START LOCATION  
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
:LOOP ON ERROR END LOCATION  
:DISABLE CACHE  
:LOCATE TEST CODE TO HIGH CACHE SPACE  
:ADDRESS OF START OF NEXT TEST

:THE FOLLOWING LOCATIONS INCLUDING 10\$  
:ARE RELOCATED TO HI CACHE SPACE

7191 043300 012700 040000  
7192 043304 012701 060000  
7193 043310 112737 000002 177750 1\$:  
7194  
7195  
7196 043316 012737 000015 177746  
7197 043324 005710  
7198 043326 005711  
7199 043330 005710  
7200 043332 005711  
7201 043334 005710  
7202 043336 005711  
7203 043340 005710  
7204 043342 013702 177752  
7205 043346 000240 25\$:  
043350 000240  
7206 043352 012737 001015 177746  
7207 043360 105037 177750  
7208 043364 042702 177700  
7209 043370 005702  
7210 043372 001406  
7211 043374 010237 047352  
7212 043400 104406

MOV #40000,R0  
MOV #60000,R1  
MOVB #HODO,CMR  
  
MOV #15,CCR  
TST (R0)  
TST (R1)  
TST (R0)  
TST (R1)  
TST (R0)  
TST (R1)  
TST (R0)  
TST (R1)  
TST (R0)  
MOV CHR,R2  
NOP  
NOP  
MOV #OFF,CCR  
CLRB CMR  
BIC #177700,R2  
TST R2  
BEQ 10\$  
MOV R2,CHR50  
ERROR

:ADDR. 40000 TO R0  
:ADDR 60000 TO R1  
:HODO ALLOWS HIT REGISTER TO BE CLOCKED  
:ONLY DURING THE DESTINATION ACCESS  
:OF AN INSTRUCTION.  
:NO UCB SO AS TO WRITE CACHE STORES  
  
:READ MISS  
:READ MISS  
:READ MISS  
:READ MISS  
:READ MISS  
:READ MISS  
:READ MISS  
:SAVE CHR CONTENTS  
:INSTRUCTION 'JMP 1\$' PLACED HERE  
:FOR LOOP ON ERROR  
:DISABLE CACHE  
:DISABLE MAINTENANCE MODE  
:PREPARE R2 FOR CHECK  
:CHR<5:0> SHOULD HAVE BEEN ALL 0'S  
:PASS  
:PREPARE FOR ERROR REPORT  
:ERROR  
:-----

043402 043400  
7213  
7214  
7215 043404 047352  
7216 043406 000000  
7217 043410 000240 10\$:  
043412 005237 001472

.WORD .-2  
  
CHR50  
.WORD 0  
NOP  
INC \$TESTN

:CHR<5:0> DID NOT INDICATE ALL 0'S  
:DUE TO SIX READ MISSES  
:PRINT CHR<5:0> RECEIVED  
  
:END OF TEST  
:INCREMENT TEST COUNTER



7222

.SBTTL TEST # 224 - BIT 05 OF HIT REG CONTAINS 1 WITH 1 HIT, 5 MISSES  
\*\*\*\*\*  
\*TEST 224 - BIT 05 OF HIT REG CONTAINS 1 WITH 1 HIT, 5 MISSES  
\* VERIFY THAT BIT 05 OF CACHE HIT REGISTER CAN CONTAIN A 1 DUE TO  
\* ONE READ HIT FOLLOWED BY FIVE READ MISSES  
\*\*\*\*\*

7223 043416 000004

043416 000004  
043420 043430  
043422 070010  
043424 000000  
043426 070046  
043430 012737 001015 177746 40\$:  
043436 004437 002370  
043442 043560

TST224:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
.WORD 40\$ ;TEST START LOCATION  
.WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION  
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION  
MOV #OFF,CCR ;DISABLE CACHE  
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE  
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

;THE FOLLOWING LOCATIONS INCLUDING 10\$  
;ARE RELOCATED TO HI CACHE SPACE

7224 043444 012700 040000  
7225 043450 012701 060000  
7226 043454 112737 000002 177750 1\$:  
7227  
7228  
7229 043462 012737 000015 177746  
7230 043470 005710  
7231 043472 005710  
7232 043474 005711  
7233 043476 005710  
7234 043500 005711  
7235 043502 005710  
7236 043504 005711  
7237 043506 013702 177752  
7238 043512 000240 25\$:  
043514 000240  
7239 043516 012737 001015 177746  
7240 043524 105037 177750  
7241 043530 042702 177700  
7242 043534 022702 000040  
7243 043540 001406  
7244 043542 010237 047352  
7245 043546 104406

MOV #40000,R0 ;ADDR. 40000 TO R0  
MOV #60000,R1 ;ADDR 60000 TO R1  
MOVB #HODO,CMR ;HODO ALLOWS HIT REGISTER TO BE CLOCKED  
;ONLY DURING THE DESTINATION ACCESS  
;OF AN INSTRUCTION.  
;NO UCB SO AS TO WRITE CACHE STORES  
MOV #15,CCR  
TST (R0)  
TST (R0) ;READ HIT  
TST (R1) ;READ MISS  
TST (R0) ;READ MISS  
TST (R1) ;READ MISS  
TST (R0) ;READ MISS  
TST (R1) ;READ MISS  
TST (R1) ;READ MISS  
MOV CHR,R2 ;SAVE CHR CONTENTS  
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
MOV #OFF,CCR ;DISABLE CACHE  
CLRB CMR ;DISABLE MAINTENANCE MODE  
BIC #177700,R2 ;PREPARE R2 FOR CHECK  
CMP #40,R2 ;BIT 05 SHOULD BE 1  
BEQ 10\$ ;PASS  
MOV R2,CHR50 ;PREPARE FOR ERROR REPORT  
ERROR ;ERROR  
;-----

043550 043546  
7246  
7247  
7248 043552 047352  
7249 043554 000000  
7250 043556 000240 10\$:  
043560 005237 001472

.WORD -2 ;CHR BIT 05 DID NOT READ 1 DUE  
;TO ONE READ HIT AND 5 READ MISSES  
;PRINT CHR<5:0> RECEIVED  
CHR50  
.WORD 0  
NOP ;END OF TEST  
INC \$TESTN ;INCREMENT TEST COUNTER

7257

```
.SBTTL TEST # 225 - EXERCISE CACHE BY READING MEMORY LOCATIONS
*****
*TEST 225 - EXERCISE CACHE BY READING MEMORY LOCATIONS
* THIS TEST EXERCISES CACHE BY READING MEMORY LOCATIONS FROM
* 60000 TO 77776 WITH CACHE ON. ALL 4K OF CACHE WILL HAVE BEEN
* EXERCISED. EACH ADDRESS FROM 60000 TO 77776 IS LOADED WITH
* DATA CORRESPONDING TO ITS OWN ADDRESS.
*****
```

```
043564
7258 043564 000004
      043566 043576
      043570 043576
      043572 000000
      043574 043706
      043576
7259 043576 012700 060000
7260 043602 010010
7261 043604 005720
7262 043606 020027 077776
7263 043612 101773
7264 043614 012700 060000
7265 043620 005037 177746
7266 043624 005110
7267 043626 005110
7268 043630 011005
7269 043632 020500
7270 043634 001420
7271 043636 012737 001015 177746
7272 043644 010037 047356
7273 043650 010037 047332
7274 043654 010537 047346
7275 043660 104406
      043662 043660
7276 043664 047356
7277 043666 047332
7278 043670 047346
7279 043672 000000
7280 043674 000404
7281 043676 005720
7282 043700 020027 077776
7283 043704 101747
7284 043706 000240
      043710 000240
7285 043712 000240
      043714 005237 001472
```

```
TST225:
      SCPCND
      .WORD 40$
      .WORD 1$
      .WORD 0
      .WORD 25$
40$:
1$: MOV #60000,R0
2$: MOV R0,(R0)
   TST (R0)+
   CMP R0,#77776
   BLOS 2$
   MOV #60000,R0
   CLR 177746
3$: COM (R0)
   COM (R0)
   MOV (R0),R5
   CMP R5,R0
   BEQ 5$
   MOV #OFF,CCR
   MOV R0,FAILAD
   MOV R0,EXDAT6
   MOV R5,RECDAT
   ERROR
      .WORD -2
      FAILAD
      EXDAT6
      RECDAT
      .WORD 0
      BR 25$
5$: TST (R0)+
   CMP R0,#77776
   BLOS 3$
25$: NOP
      NOP
10$: NOP
      INC $TESTN
;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;FIRST ADDRESS
;FILL MEMORY WITH ADDRESSES
;
;FIRST ADDRESS
;ENABLE CACHE
;DOUBLE COMPLEMENT DATA AND
;MAKE SURE IT IS IN THE CACHE
;CREATE READ HIT;STORE CACHED DATA IN R5
;CHECK RESULTS
;PASS
;DISABLE CACHE
;SAVE FAILED ADDRESS
;GET EXPECTED DATA
;GET RECEIVED DATA
;ERROR
;-----
;PRINT FAILED ADDRESS
;PRINT EXPECTED DATA
;PRINT RECEIVED DATA
;
;NEXT ADDRESS
;FINISHED?
;CONTINUE
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;END OF TEST
;INCREMENT TEST COUNTER
```

TEST # 226 - ASRB INST CAUSES CACHE BYPASS WITH READ HIT  
7291

.SBTTL TEST # 226 - ASRB INST CAUSES CACHE BYPASS WITH READ HIT  
:\*\*\*\*\*  
:TEST 226 - ASRB INST CAUSES CACHE BYPASS WITH READ HIT  
:\* TEST DESCRIPTION:  
:\* VERIFY THAT THE ASRB INSTRUCTION WILL CAUSE A CACHE BYPASS  
:\* UNDER A READ HIT CONDITION.  
:\*\*\*\*\*

7292 043720 000004  
043722 043732  
043724 043732  
043726 000000  
043730 043762  
043732  
7293 043732 012700 060000  
7294 043736 112737 000002 177750  
7295  
7296  
7297  
7298 043744 012737 000011 177746  
7299  
7300 043752 005710  
7301  
7302  
7303 043754 106210  
7304  
7305  
7306  
7307  
7308 043756 013701 177750  
7309 043762 000240  
043764 000240  
7310 043766 105037 177750  
7311 043772 012737 000000 177746  
7312 044000 032701 000400  
7313 044004 001003  
7314 044006 104406  
044010 044006  
7315  
7316  
7317  
7318  
7319 044012 000000  
7320 044014 000240  
044016 005237 001472  
7321  
7322 044022 005737 001474  
7323 044026 001412  
7324 044030 032777 004000 136036  
7325  
7326 044036 001006  
7327 044040 005237 044164  
7328 044044 023737 044162 044164  
7329 044052 001041  
7330 044054 005037 001472

TST226:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
;LOOP ON ERROR START LOCATION  
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
;LOOP ON ERROR END LOCATION  
40\$:  
1\$: MOV #60000,R0 ;SETUP TEST LOCATION ADDRESS IN R0  
MOV #2,177750 ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND  
;CLOCKING OF OUTPUT OF CACHE HIT NAND  
;GATE INTO CMR ONLY DURING THE DESTINATION  
;ACCESS OF AN INSTRUCTION.  
MOV #11,177746 ;NO UCB SO AS TO WRITE CACHE STORES  
;ENABLE LOW CACHE FOR A READ HIT  
TST (R0) ;READING LOCATION SPECIFIED BY R0  
;WILL ASSURE A READ HIT WHEN THE  
;LOCATION IS READ AGAIN  
ASRB (R0) ;ASRB INSTRUCTION WILL CAUSE A BYPASS  
;TO OCCUR INHIBITING A READ HIT  
;TO LOCATION SPECIFIED BY R0.  
;THIS SITUATION WILL RESULT IN CMR  
;BIT 8 BEING A 1.  
MOV 177750,R1 ;SAVE CMR CONTENTS  
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE  
NOP ;FOR LOOP ON ERROR  
CLR 177750 ;DISABLE MAINT MODE  
MOV #0,177746 ;TURN ON CACHE  
BIT #400,R1 ;WAS CMR BIT 8 A 1  
BNE 10\$ ;PASS  
ERROR ;ERROR  
;-----  
;CACHE BYPASS DID NOT OCCUR OR  
;SEQUENCE ERROR  
;READING OUTPUT OF CACHE HIT NAND GATE  
;THRU CMR<8> DID NOT RESULT IN A 1  
10\$: .WORD 0  
NOP ;END OF TEST  
INC \$TESTN ;INCREMENT TEST COUNTER  
:\*\*\*\*\*  
ENDPAS: TST \$PASS ;IS THIS FIRST PASS?  
BEQ 1\$ ;YES; INHIBIT ITERATIONS;  
BIT #BIT11,@SWR ;IS INHIBIT ITERATIONS IMPLEMENTED  
;THRU SWITCH REGISTER?  
BNE 1\$ ;YES  
INC \$ICNT ;INCREMENT PASS ITERATION COUNTER  
CMP \$TIMES,\$ICNT ;HAVE ALL ITERATIONS BEEN COMPLETED?  
BNE \$DOAGN ;NO,REPEAT PROGRAM  
1\$: CLR \$TESTN ;CLEAR TEST NUMBER COUNTER

```

7331 044060 005037 044164          CLR      $ICNT          ;CLEAR PASS ITERATION COUNTER
7332 044064 005237 001474          INC      $PASS         ;INCREMENT PASS COUNT
7333 044070 042737 100000 001474  BIC      #100000,$PASS  ;DON'T ALLW A NEGATIVE #
7334 044076 104401 044104          TYPE    ,65$          ;;TYPE ASCIZ STFING
      044102 000410          BR       64$          ;;GET OVER THE ASCIZ
      044104      200      105      116 65$: .ASCIZ <CRLF>/END OF PASS # /
                                          .EVEN
                                          64$:
7335 044124 013746 001474          MOV      $PASS,-(SP)   ;;SAVE $PASS FOR TYPEOUT
      044130 104405          TYPDS   ;GO TYPE--DECIMAL ASCII WITH SIGN
7336 044132 104401 002122          TYPE    , $ENULL
7337 044136 013700 000042          MOV      42,RO
7338 044142 001405          BEQ     $DOAGN
7339 044144 000005          RESET
7340 044146 004710          $ENDAD: JSR    PC,(RO)
7341 044150 000240          NOP
7342 044152 000240          NOP
7343 044154 000240          NOP
7344 044156 000137 002416          $DOAGN: JMP     BEGIN          ;START AGAIN
7345
7346 044162 000012          $TIMES: .WORD 10.
7347 044164 000000          $ICNT:  .WORD 0
  
```

```

.SBTTL TYPE ROUTINE
:*****
:*ROUTINE TO TYPE ASCIZ 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.
:*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
:*
:*CALL:
:*1) USING A TRAP INSTRUCTION
:*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
:*OR
:*      TYPE
:*      MESADR
:*
044166 105737 002111 $TYPE: TSTB $TPFLG      ;; IS THERE A TERMINAL?
044172 100002          BPL 1$              ;; BR IF YES
044174 000000          IHALT          ;; HALT HERE IF NO TERMINAL
044176 000430          BR 3$              ;; LEAVE
044200 010046          1$: MOV RO,-(SP)      ;; SAVE RO
044202 017600 000002  MOV @2(SP),RO      ;; GET ADDRESS OF ASCIZ STRING
044206 122737 000001 001506  CMPB #APTENV,$ENV      ;; RUNNING IN APT MODE
044214 001011          BNE 62$          ;; NO,GO CHECK FOR APT CONSOLE
044216 132737 000100 001507  BITB #APTPOOL,$ENVM    ;; SPOOL MESSAGE TO APT
044224 001405          BEQ 62$          ;; NO,GO CHECK FOR CONSOLE
044226 010037 044236  MOV RO,61$          ;; SETUP MESSAGE ADDRESS FOR APT
044232 004737 001620  JSR PC,$ATY3      ;; SPOOL MESSAGE TO APT
044236 000000          61$: .WORD 0      ;; MESSAGE ADDRESS
044240 132737 000040 001507  62$: BITB #APTCSUP,$ENVM    ;; APT CONSOLE SUPPRESSED
044246 001003          BNE 60$          ;; YES,SKIP TYPE OUT
044250 112046          2$: MOVB (RO)+,-(SP)    ;; PUSH CHARACTER TO BE TYPED ONTO STACK
044252 001005          BNE 4$              ;; BR IF IT ISN'T THE TERMINATOR
044254 005726          TST (SP)+          ;; IF TERMINATOR POP IT OFF THE STACK
044256 012600          60$: MOV (SP)+,RO      ;; RESTORE RO
044260 062716 000002  3$: ADD #2,(SP)      ;; ADJUST RETURN PC
044264 000002          RTI              ;; RETURN
044266 122716 000011  4$: CMPB #HT,(SP)      ;; BRANCH IF <HT>
044272 001430          BEQ 8$              ;;
044274 122716 000200  CMPB #CRLF,(SP)      ;; BRANCH IF NOT <CRLF>
044300 001006          BNE 5$              ;;
044302 005726          TST (SP)+          ;; POP <CR><LF> EQUIV
044304 104401          TYPE          ;; TYPE A CR AND LF
044306 002117          $CRLF
044310 105037 044514  CLRB $CHARCNT      ;; CLEAR CHARACTER COUNT
044314 000755          BR 2$              ;; GET NEXT CHARACTER
044316 004737 044400  5$: JSR PC,$TYPEC      ;; GO TYPE THIS CHARACTER
044322 123726 002110  6$: CMPB $FILLC,(SP)+    ;; IS IT TIME FOR FILLER CHARS.?
044326 001350          BNE 2$              ;; IF NO GO GET NEXT CHAR.
044330 013746 002106  MOV $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
                                ;; AND THE NULL CHAR.
044334 105366 000001  7$: DECB 1(SP)          ;; DOES A NULL NEED TO BE TYPED?
044340 002770          BLT 6$              ;; BR IF NO--GO POP THE NULL OFF OF STACK
044342 004737 044400  JSR PC,$TYPEC      ;; GO TYPE A NULL
044346 105337 044514  DECB $CHARCNT      ;; DO NOT COUNT AS A COUNT
044352 000770          BR 7$              ;; LOOP
:HORIZONTAL TAB PPROCESSOR
044354 112716 000040  8$: MOVB #' ,(SP)      ;; REPLACE TAB WITH SPACE

```

```

044360 004737 044400          9$:   JSR      PC,$TYPEC      ;;TYPE A SPACE
044364 132737 000007 044514   BITB    #7,$CHARCNT    ;;BRANCH IF NOT AT
044372 001372          BNE     9$             ;;TAB STOP
044374 005726          TST     (SP)+          ;;POP SPACE OFF STACK
044376 000724          BR      2$             ;;GET NEXT CHARACTER
044400 105777 135476          $TYPEC: TSTB   @STPS      ;;WAIT UNTIL PRINTER IS READY
044404 100375          BPL     $TYPEC
044406 116677 000002 135470   MOVB    2(SP),@STPB    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
044414 105777 135456          TSTB   @STKS          ;;SEE IF KEYBOARD IS TALKING.
044420 100021          BPL     2$             ;;BRANCH IF IT ISN'T.
044422 017746 135452          MOV     @STKB,-(SP)    ;;PUSH CHARACTER ONTO STACK.
044426 042716 177600          BIC     #177600,(SP)  ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044432 022726 000023          CMP     #23,(SP)+     ;;SEE IF THIS IS A ^S.
044436 001012          BNE     2$             ;;BRANCH TO CONTINUE IF IT ISN'T.
044440 105777 135432          3$:   TSTB   @STKS          ;;WAIT FOR ANOTHER INPUT.
044444 100375          BPL     3$             ;;BRANCH BACK IF NOT READY.
044446 017746 135426          MOV     @STKB,-(SP)  ;;PUSH NEXT CHARACTER ON STACK.
044452 042716 177600          BIC     #177600,(SP)  ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044456 022726 000021          CMP     #21,(SP)+     ;;SEE IF THIS IS A ^Q.
044462 001366          BNE     3$             ;;BRANCH BACK FOR MORE WAIT IF NOT.
044464 122766 000015 000002  2$:   CMPB   #CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
044472 001003          BNE     1$             ;;BRANCH IF NO
044474 105037 044514          CLRB   $CHARCNT      ;;YES--CLEAR CHARACTER COUNT
044500 000406          BR     $TYPEX        ;;EXIT
044502 122766 000012 000002  1$:   CMPB   #LF,2(SP)     ;;IS CHARACTER A LINE FEED?
044510 001402          BEQ    $TYPEX        ;;BRANCH IF YES
044512 105227          INCB   (PC)+         ;;COUNT THE CHARACTER
044514 000000          $CHARCNT: .WORD    0 ;;CHARACTER COUNT STORAGE
044516 000207          $TYPEX: RTS        PC

```

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE
*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS    ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*
*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    ;;CALL FOR TYPEOUT
044520 017646 000000      $TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
044524 116637 000001 044743  MOV      1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
044532 112637 044745      MOV      (SP)+, $OMODE+1    ;;NUMBER OF DIGITS TO TYPE
044536 062716 000002      ADD      #2, (SP)          ;;ADJUST RETURN ADDRESS
044542 000406      BR      $TYPON
044544 112737 000001 044743 $TYPOC: MOV      #1, $OFILL      ;;SET THE ZERO FILL SWITCH
044552 112737 000006 044745  MOV      #6, $OMODE+1      ;;SET FOR SIX(6) DIGITS
044560 112737 000005 044742 $TYPON: MOV      #5, $OCNT      ;;SET THE ITERATION COUNT
044566 010346      MOV      R3, -(SP)        ;;SAVE R3
044570 010446      MOV      R4, -(SP)        ;;SAVE R4
044572 010546      MOV      R5, -(SP)        ;;SAVE R5
044574 113704 044745      MOV      $OMODE+1, R4     ;;GET THE NUMBER OF DIGITS TO TYPE
044600 005404      NEG      R4
044602 062704 000006      ADD      #6, R4           ;;SUBTRACT IT FOR MAX. ALLOWED
044606 110437 044744      MOV      R4, $OMODE      ;;SAVE IT FOR USE
044612 113704 044743      MOV      $OFILL, R4      ;;GET THE ZERO FILL SWITCH
044616 016605 000012      MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
044622 005003      CLR      R3              ;;CLEAR THE OUTPUT WORD
044624 006105      1$: ROL      R5           ;;ROTATE MSB INTO 'C'
044626 000404      BR      3$              ;;GO DO MSB
044630 006105      2$: ROL      R5           ;;FORM THIS DIGIT
044632 006105      ROL      R5
044634 006105      ROL      R5
044636 010503      MOV      R5, R3
044640 006103      3$: ROL      R3           ;;GET LSR OF THIS DIGIT
044642 105337 044744      DECB    $OMODE          ;;TYPE THIS DIGIT?
044646 100016      BPL      7$              ;;BR IF NO
044650 042703 177770      BIC      #177770, R3     ;;GET RID OF JUNK
044654 001002      BNE      4$              ;;TEST FOR 0
044656 005704      TST     R4              ;;SUPPRESS THIS 0?
044660 001403      BEQ     5$              ;;BR IF YES
044662 005204      4$: INC      R4          ;;DON'T SUPPRESS ANYMORE 0'S
044664 052703 000060      BIS     #'0, R3         ;;MAKE THIS DIGIT ASCII
044670 052703 000040      5$: BIS     #' ,R3       ;;MAKE ASCII IF NOT ALREADY

```

```

044674 110337 044740      MOVB   R3,B$      ;;SAVE FOR TYPING
044700 104401 044740      TYPE   ,B$       ;;GO TYPE THIS DIGIT
044704 105337 044742      7$:   DECB   $OCNT  ;;COUNT BY 1
044710 003347           BGT    2$        ;;BR IF MORE TO DO
044712 002402           BLT    6$        ;;BR IF DONE
044714 005204           INC    R4        ;;INSURE LAST DIGIT ISN'T A BLANK
044716 000744           BR     2$        ;;GO DO THE LAST DIGIT
044720 012605           6$:   MOV   (SP)+,R5  ;;RESTORE R5
044722 012604           MOV   (SP)+,R4  ;;RESTORE R4
044724 012603           MOV   (SP)+,R3  ;;RESTORE R3
044726 016666 000002 000004  MOV   2(SP),4(SP) ;;SET THE STACK FOR RETURNING
044734 012616           MOV   (SP)+,(SP)
044736 000002           RTI                    ;;RETURN
044740      000      8$:   .BYTE  0      ;;STORAGE FOR ASCII DIGIT
044741      000           .BYTE  0      ;;TERMINATOR FOR TYPE ROUTINE
044742      000      $OCNT: .BYTE  0      ;;OCTAL DIGIT COUNTER
044743      000      $OFILL: .BYTE  0      ;;ZERO FILL SWITCH
044744 000000      $CMODE: .WORD  0      ;;NUMBER OF DIGITS TO TYPE

```



```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
:*****
:*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
:*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
:*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
:*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
:*REPLACED WITH SPACES.
:*CALL:
:*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
:*      TYPDS      ;;GO TO THE ROUTINE
$TYPDS:
MOV      R0,-(SP)      ;;PUSH R0 ON STACK
MOV      R1,-(SP)      ;;PUSH R1 ON STACK
MOV      R2,-(SP)      ;;PUSH R2 ON STACK
MOV      R3,-(SP)      ;;PUSH R3 ON STACK
MOV      R5,-(SP)      ;;PUSH R5 ON STACK
MOV      #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
MOV      20(SP),R5     ;;GET THE INPUT NUMBER
BPL      1$           ;;BR IF INPUT IS POS.
NEG      R5           ;;MAKE THE BINARY NUMBER POS.
MOVB     #'-,1(SP)    ;;MAKE THE ASCII NUMBER NEG.
1$:      CLR      R0      ;;ZERO THE CONSTANTS INDEX
MOV      #5DBLK,R3    ;;SETUP THE OUTPUT POINTER
MOVB     #' ,(R3)+    ;;SET THE FIRST CHARACTER TO A BLANK
2$:      CLR      R2      ;;CLEAR THE BCD NUMBER
MOV      $DTBL(R0),R1 ;;GET THE CONSTANT
3$:      SUB      R1,R5    ;;FORM THIS BCD DIGIT
BLT      4$           ;;BR IF DONE
INC      R2           ;;INCREASE THE BCD DIGIT BY 1
BR       3$
4$:      ADD      R1,R5    ;;ADD BACK THE CONSTANT
TST      R2           ;;CHECK IF BCD DIGIT=0
BNE      5$           ;;FALL THROUGH IF 0
TSTB     (SP)         ;;STILL DOING LEADING 0'S?
BMI      7$           ;;BR IF YES
5$:      ASLB     (SP)     ;;MSD?
BCC      6$           ;;BR IF NO
MOVB     1(SP),-1(R3) ;;YES--SET THE SIGN
6$:      BIS      #'0,R2   ;;MAKE THE BCD DIGIT ASCII
7$:      BIS      #' ,R2   ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB     R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST      (R0)+       ;;JUST INCREMENTING
CMP      R0,#10      ;;CHECK THE TABLE INDEX
BLT      2$           ;;GO DO THE NEXT DIGIT
BGT      8$           ;;GO TO EXIT
MOV      R5,R2       ;;GET THE LSD
BR       6$           ;;GO CHANGE TO ASCII
8$:      TSTB     (SP)+    ;;WAS THE LSD THE FIRST NON-ZERO?
BPL      9$           ;;BR IF NO
MOVB     -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
9$:      CLRB     (R3)     ;;SET THE TERMINATOR
MOV      (SP)+,R5     ;;POP STACK INTO R5
MOV      (SP)+,R3     ;;POP STACK INTO R3
MOV      (SP)+,R2     ;;POP STACK INTO R2
MOV      (SP)+,R1     ;;POP STACK INTO R1
MOV      (SP)+,R0     ;;POP STACK INTO R0
TYPE     ,5DBLK      ;;NOW TYPE THE NUMBER

```

```

044746
044746 010046
044750 010146
044752 010246
044754 010346
044756 010546
044760 012746 0202C0
044764 016605 000020
044770 100004
044772 005405
044774 112766 000055 000001
045002 005000
045004 012703 045162
045010 112723 000040
045014 005002
045016 016001 045152
045022 160105
045024 002402
045026 005202
045030 000774
045032 060105
045034 005702
045036 001002
045040 105716
045042 100407
045044 106316
045046 103003
045050 116663 000001 177777
045056 052702 000060
045062 052702 000040
045066 110223
045070 005720
045072 020027 000010
045076 002746
045100 003002
045102 010502
045104 000764
045106 105726
045110 100003
045112 116663 177777 177776
045120 105013
045122 012605
045124 012603
045126 012602
045130 012601
045132 012600
045134 104401 045162

```

```
045140 016666 000002 000004      MOV      2(SP),4(SP)      ;;ADJUST THE STACK
045146 012616                      MOV      (SP)+,(SP)
045150 000002                      RTI                      ;;RETUPN TO USER
045152 023420                      $DTBL: 10000.
045154 001750                                           1000.
045156 000144                                           100.
045160 000012                                           10.
045162                      $DBLK: .BLKW 4
```

7354

045172 010046  
045174 016600 000002  
045200 005740  
045202 111000  
045204 006300  
045206 016000 045226  
045212 000200

045214 011646  
045216 016666 000004 000002  
045224 000002

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

```
$TRAP:  MOV    R0,-(SP)           ;;SAVE R0
        MOV    2(SP),R0         ;;GET TRAP ADDRESS
        TST   -(R0)             ;;BACKUP BY 2
        MOVB  (R0),R0           ;;GET RIGHT BYTE OF TRAP
        ASL   R0                 ;;POSITION FOR INDEXING
        MOV   $TRPAD(R0),R0     ;;INDEX TO TABLE
        RTS   R0                 ;;GO TO ROUTINE
```

```
;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
$TRAP2: MOV   (SP),-(SP)        ;;MOVE THE PC DOWN
        MOV   4(SP),2(SP)      ;;MOVE THE PSW DOWN
        RTI                      ;;RESTORE THE PSW
```

```
.SBTTL TRAP TABLE
*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
*BY THE 'TRAP' INSTRUCTION.
ROUTINE
```

045226 045214  
045230 044166  
045232 044544  
045234 044520  
045236 044560  
045240 044746  
7355 045242 045246

```
$TRPAD: .WORD  $TRAP2
        $TYPE  ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
        $TYPOC ;;CALL=TYPOC    TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS ;;CALL=TYPOS    TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON ;;CALL=TYPON    TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS ;;CALL=T:PDS    TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
        $SERPOR ;;CALL=ERROR   TRAP+6(104406)
```

7357

7358

7359 045244 000000  
7360 045246 010037 045300  
7361 045252 010137 045302  
7362 045256 010237 045304  
7363 045262 010337 045306  
7364 045266 010437 045310  
7365 045272 010537 045312  
7366 045276 000406  
7367 045300 000000  
7368 045302 000000  
7369 045304 000000  
7370 045306 000000  
7371 045310 000000  
7372 045312 000000

7373

7374 045314 113777 002060 134552  
7375 045322 032777 020000 134544  
7376 045330 001113  
7377 045332 011601  
7378 045334 012137 045244  
7379  
7380 045340 104401 045346  
045344 000411  
045346 200 200 124 65\$:

045370

7381 045370 005711  
7382 045372 001412  
7383 045374 012702 047246  
7384 045400 012703 045752  
7385 045404 005723  
7386 045406 021122  
7387 045410 001375  
7388 045412 004753  
7389 045414 005721  
7390 045416 000764  
7391 045420

045420 104401 045426  
045424 000401  
045426 200 000 65\$:

045430

7392 045430 013746 001472  
045434 104403  
045436 006  
045437 000  
7393 045440 104401 045446  
045444 000402  
045446 040 040 000 67\$:

045452

7394 045452 013746 045244  
045456 104402  
7395 045460 104401 002122  
7396 045464 011601

.SBTTL ERROR HANDLER ROUTINE

\*\*\*\*\*

```

ERRPC: .WORD 0
$ERROR: MOV R0,SAVR0 ;SAVE R0 THRU R5
        MOV R1,SAVR1
        MOV R2,SAVR2
        MOV R3,SAVR3
        MOV R4,SAVR4
        MOV R5,SAVR5
        BR TITL

SAVR0: .WORD
SAVR1: .WORD
SAVR2: .WORD
SAVR3: .WORD
SAVR4: .WORD
SAVR5: .WORD

TITL:   MOVB $STNM,@SWR ;MOVE TEST NUMBER TO DISPLAY FOR ALL TO SEE
        BIT #BIT13,@SWR ;INHIBIT ERROR TYPEOUTS?
        BNE INHERR ;YES
        MOV (SP),R1 ;R1 CONTAINS ADDRESS FOLLOWING ERRPC ADDRESS
        MOV (R1)+,ERRPC ;LOAD ERRPC ADDRESS AND POINT
        ;TO NEXT ARGUMENT
        TYPE ,65$ ;:TYPE ASCIZ STRING
        BR 64$ ;:GET OVER THE ASCIZ
65$:   .ASCIZ <CRLF><CRLF>/TESTNO ERRPC/
        .EVEN

64$:   TST (R1) ;END OF ARGUMENTS?
3$:   BEQ DAT ;YES,GO PRINT DATA
        MOV #PRTABL,R2 ;ADDRESS OF START OF PRINT TABLE LIST
        MOV #PRTITL,R3 ;ADDRESS OF START OF TITLES
2$:   TST (R3)+ ;INDEX THRU TITLES
        CMP (R1),(R2)+ ;SEARCH PRINT TABLE LIST FOR TITLE
        BNE 2$ ;NO; CHECK NEXT LOCATION IN LIST
        JSR PC,@-(R3) ;FOUND IT; GO PRINT TITLE
        TST (R1)+ ;R1 POINTS TO NXT ARGUMENT IN TEST CODE
        BR 3$

DAT:   TYPE ,65$ ;:TYPE ASCIZ STRING
        BR 64$ ;:GET OVER THE ASCIZ
65$:   .ASCIZ <CRLF>
        .EVEN

64$:   MOV $TESTN,-(SP) ;:SAVE $TESTN FOR TYPEOUT
        TYPOS ;:GO TYPE--OCTAL ASCII
        .BYTE 6 ;:TYPE 6 DIGIT(S)
        .BYTE 0 ;:SUPPRESS LEADING ZEROS
        TYPE ,67$ ;:TYPE ASCIZ STRING
        BR 66$ ;:GET OVER THE ASCIZ
67$:   .ASCIZ / /
        .EVEN

66$:   MOV ERRPC,-(SP) ;:SAVE ERRPC FOR TYPEOUT
        TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
        TYPE , $ENULL
        MOV (SP),R1 ;R1 CONTAINS ADDRESS FOLLOWING ERRORPC ADDRESS
    
```

```

7397 045466 005721          TST      (R1)+          ;POINT TO NEXT ARGUEMENT
7398 045470 005711          TST      (R1)           ;END OF ARGUEMENTS?
7399 045472 001437          BEQ      FIN            ;YES
7400 045474 012702 047246    MOV      #PRTABL,R2     ;ADDRESS OF START OF PRINT TABLE LIST
7401 045500 012703 046724    MOV      #PRDATA,R3    ;ADDRESS OF STERT OF PRINT DATA
7402 045504 005723          TST      (R3)+          ;INDEX THRU DATA PRINTS
7403 045506 021122          CMP      (R1),(R2)+    ;SEARCH PRINT TABLE LIST FOR TITLE
7404 045510 001375          BNE     2$             ;NO; CHECK NEXT LOCATION IN LIST
7405 045512 104401 045520    TYPE    ,69$          ;:TYPE ASCIZ STRING
      045516 000404          BR       68$          ;:GET OVER THE ASCIZ
      045520      040      040 69$: .ASCIZ / /
      .EVEN
      68$:
7406 045530 004753          JSR     PC,@-(R3)      ;
7407 045532 104401 002122    TYPE    , $NULL
7408 045536 104401 045544    TYPE    ,71$          ;:TYPE ASCIZ STRING
      045542 000404          BR       70$          ;:GET OVER THE ASCIZ
      045544      040      040 71$: .ASCIZ / /
      .EVEN
      70$:
7409 045554 005721          TST      (R1)+          ;R1 POINTS TO NEXT ARGUEMENT
7410 045556 000744          BR       3$           ;INHIBIT ERROR TYPEOUT CODE
7411
7412
7413 045560 011601          INHERR: MOV (SP),R1     ;R1 CONTAINS ADDRESS FOLLOWING ERRPC ADDRESS
7414 045562 005711          1$:    TST      (R1)           ;IS THIS THE END OF ARGUEMENT LIST?
7415 045564 001402          BEQ      FIN            ;YES
7416 045566 005721          TST      (R1)+          ;POINT TO NEXT ARGUEMENT
7417 045570 000774          BR       1$            ;
7418 045572 005721          FIN:   TST      (R1)+          ;R1 NOW CONTAINS RETURN ADDRESS
7419 045574 010116          MOV      R1,(SP)       ;SETUP RETURN ADDRESS IN STACK
7420
7421
7422
      ;HOE,LOE OPTION DEIERMINATION
7423 045576 122737 000001 001506    CMPB    #APTENV,$ENV   ;IS THIS APT?
7424 045604 001410          BEQ     2$             ;YES HALT ON ERROR
7425 045606 023737 000042 000046    CMP     42,46          ;IS THIS ACT11 QV OR AUTO ACCEPT
7426 045614 001404          BEQ     2$             ;YES HALT ON ERROR
7427 045616 032777 100000 134250    BIT     #BIT15,@SWR    ;IS HALT ON ERROR IMPLEMENTED?
7428 045624 001404          BEQ     1$            ;NO
7429 045626 012737 000001 001466    2$:    MOV     #1,$MSGTY   ;SET $MSGTY FOR POSSIBLE APT USE
7430 045634 000000          HALT
7431 045636 032777 001000 134230    1$:    BIT     #BIT09,@SWR  ;IS LOOP ON ERROR IMPLEMENTED?
7432 045644 001010          BNE     4$            ;YES
7433
7434
7435
      ;NO; INITIALIZE LOCATIONS WHERE
      ;'JMP 1$' IS PLACED FOR LOOP ON ERROR
      ;WITH NOP'S
7436 045646 012777 000240 134260    MOV     #240,@ADRJMP
7437 045654 012777 000240 134254    MOV     #240,@ADR1$
7438 045662 000137 045720          JMP     5$            ;CONTINUE WITH PRESENT TEST
7439 045666 012777 000137 134240    4$:    MOV     #137,@ADRJMP ;WRITE 'JMP' INSTRUCTION TO PROPER ADDRESS
7440 045674 013777 002130 134234    MOV     STRLTP,@ADR1$ ;WRITE '1$' LOCATION TO PROPER ADDRESS
7441 045702 013737 002132 177752    MOV     ADRSYNC,CHR    ;LOAD ADDRESS LOCATION FOR SCOPE SYNC
7442 045710 105037 177751    CLRB   CMR+1
7443 045714 013716 002130          MOV     STRLTP,(SP)   ;SETUP LOCATION FOR LOOP ON ERROR IN STACK
7444 045720 013700 045300    5$:    MOV     SAVR0,R0      ;RESTORE REGISTERS
7445 045724 013701 045302    MOV     SAVR1,R1

```

CKKKABO 11-44 KK11B CACHE  
ERROR HANDLER ROUTINE

MACRO M1113 27-OCT-80 08:39 PAGE 176-2 SEQUENCE 276

D 6

7446 045730 013702 045304  
7447 045734 013703 045306  
7448 045740 013704 045310  
7449 045744 013705 045312  
7450 045750 000002

MOV SAVR2,R2  
MOV SAVR3,R3  
MOV SAVR4,R4  
MOV SAVR5,R5  
RTI

;RETURN

CKKKABO 11-44 KK11B CACHE  
ERROR HANDLER ROUTINE

MACRO M1113 27-OCT-80 08:39 PAGE 177 SEQUENCE 277

```

7451 045752 046014 046050 046104 PRTITL: .WORD 1$,2$,3$,4$,5$,6$,7$,8$,9$,10$,11$,12$,13$,14$,15$,16$,17$
7452 046014 1$:
      046014 104401 046022      TYPE      ,65$      ;;TYPE ASCIZ STRING
      046020 000412      BR      ,64$      ;;GET OVER THE ASCIZ
      046022 040 040 040 65$: .ASCIZ / CA210(21:0) /
      .EVEN
      64$:
7453 046046 000207      RTS PC
7454 046050 2$:
      046050 104401 046056      TYPE      ,67$      ;;TYPE ASCIZ STRING
      046054 000412      BR      ,66$      ;;GET OVER THE ASCIZ
      046056 040 040 040 67$: .ASCIZ / AMR210(21:0) /
      .EVEN
      66$:
7455 046102 000207      RTS PC
7456 046104 3$:
      046104 104401 046112      TYPE      ,69$      ;;TYPE ASCIZ STRING
      046110 000412      BR      ,68$      ;;GET OVER THE ASCIZ
      046112 040 040 040 69$: .ASCIZ / CHR157(15:07) /
      .EVEN
      68$:
7457 046136 000207      RTS PC
7458 046140 4$:
      046140 104401 046146      TYPE      ,71$      ;;TYPE ASCIZ STRING
      046144 000412      BR      ,70$      ;;GET OVER THE ASCIZ
      046146 040 040 040 71$: .ASCIZ / CA2113(21:13) /
      .EVEN
      70$:
7459 046172 000207      RTS PC
7460 046174 5$:
      046174 104401 046202      TYPE      ,73$      ;;TYPE ASCIZ STRING
      046200 000412      BR      ,72$      ;;GET OVER THE ASCIZ
      046202 040 040 040 73$: .ASCIZ / CHR80(8:0) /
      .EVEN
      72$:
7461 046226 000207      RTS PC
7462 046230 6$:
      046230 104401 046236      TYPE      ,75$      ;;TYPE ASCIZ STRING
      046234 000412      BR      ,74$      ;;GET OVER THE ASCIZ
      046236 040 040 040 75$: .ASCIZ / CDR150(15:0) /
      .EVEN
      74$:
7463 046262 000207      RTS PC
7464 046264 7$:
      046264 104401 046272      TYPE      ,77$      ;;TYPE ASCIZ STRING
      046270 000412      BR      ,76$      ;;GET OVER THE ASCIZ
      046272 040 040 040 77$: .ASCIZ / EXDAT6 /
      .EVEN
      76$:
7465 046316 000207      RTS PC
7466 046320 8$:
      046320 104401 046326      TYPE      ,79$      ;;TYPE ASCIZ STRING
      046324 000412      BR      ,78$      ;;GET OVER THE ASCIZ
      046326 040 040 040 79$: .ASCIZ / CA121(12:1) /
      .EVEN
      78$:
7467 046352 000207      RTS PC

```

CKKKABO 11-44 KK11B CACHE  
ERROR HANDLER ROUTINE

7468	046354				9\$:					
	046354	104401	046362			TYPE	81\$		::TYPE ASCIZ STRING	
	046360	000412				BR	80\$		::GET OVER THE ASCIZ	
	046362	040	040	040	81\$:	.ASCIZ	/	EXDAT1	/	
						.EVEN				
	046406				80\$:					
7469	046406	000207				RTS PC				
7470	046410				10\$:					
	046410	104401	046416			TYPE	83\$		::TYPE ASCIZ STRING	
	046414	000412				BR	82\$		::GET OVER THE ASCIZ	
	046416	040	040	040	83\$:	.ASCIZ	/	CM1513(15:13)	/	
						.EVEN				
	046442				82\$:					
7471	046442	000207				RTS PC				
7472	046444				11\$:					
	046444	104401	046452			TYPE	85\$		::TYPE ASCIZ STRING	
	046450	000412				BR	84\$		::GET OVER THE ASCIZ	
	046452	040	040	040	85\$:	.ASCIZ	/	CNT121(12:1)	/	
						.EVEN				
	046476				84\$:					
7473	046476	000207				RTS PC				
7474	046500				12\$:					
	046500	104401	046506			TYPE	87\$		::TYPE ASCIZ STRING	
	046504	000412				BR	86\$		::GET OVER THE ASCIZ	
	046506	040	040	040	87\$:	.ASCIZ	/	FLTPAT	/	
						.EVEN				
	046532				86\$:					
7475	046532	000207				RTS PC				
7476	046534				13\$:					
	046534	104401	046542			TYPE	89\$		::TYPE ASCIZ STRING	
	046540	000406				BR	88\$		::GET OVER THE ASCIZ	
	046542	011	122	105	89\$:	.ASCIZ	/	RECDAT	/	
						.EVEN				
	046556				88\$:					
7477	046556	000207				RTS PC				
7478	046560				14\$:					
	046560	104401	046566			TYPE	91\$		::TYPE ASCIZ STRING	
	046564	000411				BR	90\$		::GET OVER THE ASCIZ	
	046566	040	040	040	91\$:	.ASCIZ	/	EXDAT3	/	
						.EVEN				
	046610				90\$:					
7479	046610	000207				RTS PC				
7480	046612				15\$:					
	046612	104401	046620			TYPE	93\$		::TYPE ASCIZ STRING	
	046616	000407				BR	92\$		::GET OVER THE ASCIZ	
	046620	011	103	110	93\$:	.ASCIZ	/	CHR50	/	
						.EVEN				
	046636				92\$:					
7481	046636	000207				RTS PC				
7482	046640				16\$:					
	046640	104401	046646			TYPE	95\$		::TYPE ASCIZ STRING	
	046644	000411				BR	94\$		::GET OVER THE ASCIZ	
	046646	040	040	040	95\$:	.ASCIZ	/	CMR119	/	
						.EVEN				
	046670				94\$:					
7483	046670	000207				RTS PC				
7484	046672				17\$:					



CKKKABO 11-44 KK11B CACHE  
ERROR HANDLER ROUTINE

MACRO M1113 27-OCT-80 08:39 PAGE 177-2 SEQUENCE 279

G 6

046672	104401	046700			TYPE	.97\$		::TYPE ASCIZ STRING
046676	000411				BR	.96\$		:::GET OVER THE ASCIZ
046700	040	040	040	97\$:	.ASCIZ	/	FAILAD	/
					.EVEN			
046722				96\$:				
7485	046722	000207			RTS PC			

```

7486 046724 046766 047010 047032 PRDATA: .WORD 1$,2$,3$,4$,5$,6$,7$,8$,9$,10$,11$,12$,13$,14$,15$,16$,17$
7487 046766 1$:
046766 013746 047312 MOV CA210,-(SP) ;;SAVE CA210 FOR TYPEOUT
046772 104403 TYPOS ;;GO TYPE--OCTAL ASCII
046774 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
046775 001 .BYTE 1 ;;TYPE LEADING ZEROS
7488 046776 013746 047314 MOV CA210+2,-(SP) ;;SAVE CA210+2 FOR TYPEOUT
047002 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047004 005 .BYTE 5 ;;TYPE 5 DIGIT(S)
047005 001 .BYTE 1 ;;TYPE LEADING ZEROS
7489 047006 000207 RTS PC
7490 047010 2$:
047010 013746 047316 MOV AMR210,-(SP) ;;SAVE AMR210 FOR TYPEOUT
047014 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047016 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
047017 001 .BYTE 1 ;;TYPE LEADING ZEROS
7491 047020 013746 047320 MOV AMR210+2,-(SP) ;;SAVE AMR210+2 FOR TYPEOUT
047024 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047026 005 .BYTE 5 ;;TYPE 5 DIGIT(S)
047027 001 .BYTE 1 ;;TYPE LEADING ZEROS
7492 047030 000207 RTS PC
7493 047032 3$:
047032 013746 047322 MOV CHR157,-(SP) ;;SAVE CHR157 FOR TYPEOUT
047036 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047040 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
047041 001 .BYTE 1 ;;TYPE LEADING ZEROS
7494 047042 000207 RTS PC
7495 047044 4$:
047044 013746 047324 MOV CA2113,-(SP) ;;SAVE CA2113 FOR TYPEOUT
047050 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047052 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
047053 001 .BYTE 1 ;;TYPE LEADING ZEROS
7496 047054 000207 RTS PC
7497 047056 5$:
047056 013746 047326 MOV CHR80,-(SP) ;;SAVE CHR80 FOR TYPEOUT
047062 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047064 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
047065 001 .BYTE 1 ;;TYPE LEADING ZEROS
7498 047066 000207 RTS PC
7499 047070 6$:
047070 013746 047330 MOV CDR150,-(SP) ;;SAVE CDR150 FOR TYPEOUT
047074 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7500 047076 000207 RTS PC
7501 047100 7$:
047100 013746 047332 MOV EXDAT6,-(SP) ;;SAVE EXDAT6 FOR TYPEOUT
047104 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7502 047106 000207 RTS PC
7503 047110 8$:
047110 013746 047334 MOV CA121,-(SP) ;;SAVE CA121 FOR TYPEOUT
047114 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047116 004 .BYTE 4 ;;TYPE 4 DIGIT(S)
047117 001 .BYTE 1 ;;TYPE LEADING ZEROS
7504 047120 000207 RTS PC
7505 047122 9$:
047122 013746 047336 MOV EXDAT1,-(SP) ;;SAVE EXDAT1 FOR TYPEOUT
047126 104403 TYPOS ;;GO TYPE--OCTAL ASCII
047130 001 .BYTE 1 ;;TYPE 1 DIGIT(S)
  
```

	047131	001		.BYTE	1	::TYPE LEADING ZEROS
7506	047132	000207		RTS PC		
7507	047134		10\$:			
	047134	013746	047340	MOV	CM1513,-(SP)	::SAVE CM1513 FOR TYPEOUT
	047140	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047142	001		.BYTE	1	::TYPE 1 DIGIT(S)
	047143	001		.BYTE	1	::TYPE LEADING ZEROS
7508	047144	000207		RTS PC		
7509	047146		11\$:			
	047146	013746	047342	MOV	CNT121,-(SP)	::SAVE CNT121 FOR TYPEOUT
	047152	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047154	004		.BYTE	4	::TYPE 4 DIGIT(S)
	047155	001		.BYTE	1	::TYPE LEADING ZEROS
7510	047156	000207		RTS PC		
7511	047160		12\$:			
	047160	013746	047344	MOV	FLTPAT,-(SP)	::SAVE FLTPAT FOR TYPEOUT
	047164	104402		TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
7512	047166	000207		RTS PC		
7513	047170		13\$:			
	047170	013746	047346	MOV	RECDAT,-(SP)	::SAVE RECDAT FOR TYPEOUT
	047174	104402		TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
7514	047176	000207		RTS PC		
7515	047200		14\$:			
	047200	013746	047350	MOV	EXDAT3,-(SP)	::SAVE EXDAT3 FOR TYPEOUT
	047204	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047206	003		.BYTE	3	::TYPE 3 DIGIT(S)
	047207	001		.BYTE	1	::TYPE LEADING ZEROS
7516	047210	000207		RTS PC		
7517	047212		15\$:			
	047212	013746	047352	MOV	CHR50,-(SP)	::SAVE CHR50 FOR TYPEOUT
	047216	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047220	002		.BYTE	2	::TYPE 2 DIGIT(S)
	047221	001		.BYTE	1	::TYPE LEADING ZEROS
7518	047222	000207		RTS PC		
7519	047224		16\$:			
	047224	013746	047354	MOV	CMR119,-(SP)	::SAVE CMR119 FOR TYPEOUT
	047230	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047232	001		.BYTE	1	::TYPE 1 DIGIT(S)
	047233	001		.BYTE	1	::TYPE LEADING ZEROS
7520	047234	000207		RTS PC		
7521	047236		17\$:			
	047236	013746	047356	MOV	FAILAD,-(SP)	::SAVE FAILAD FOR TYPEOUT
	047242	104402		TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
7522	047244	000207		RTS PC		

```
7523 047246 047312 000000 047316 PRIBL: .WORD CA210, AMR210, CHR157, CA2113, CHR80, CDR150, EXDAT6, CA121, EXDAT1
7524 047272 047340 047342 047344 .WORD CM1513, CNT121, FLTPAT, RECDAT, EXDAT3, CHR50, CMR119, FAILAD
7525
7526 047312 000000 000000 CA210: .WORD 0,0
7527 047316 000000 000000 AMR210: .WORD 0,0
7528 047322 000000 CHR157: .WORD 0
7529 047324 000000 CA2113: .WORD 0
7530 047326 000000 CHR80: .WORD 0
7531 047330 000000 CDR150: .WORD 0
7532 047332 000000 EXDAT6: .WORD 0
7533 047334 000000 CA121: .WORD 0
7534 047336 000000 EXDAT1: .WORD 0
7535 047340 000000 CM1513: .WORD 0
7536 047342 000000 CNT121: .WORD 0
7537 047344 000000 FLTPAT: .WORD 0
7538 047346 000000 RECDAT: .WORD 0
7539 047350 000000 EXDAT3: .WORD 0
7540 047352 000000 CHR50: .WORD 0
7541 047354 000000 CMR119: .WORD 0
7542 047356 000000 FAILAD: .WORD 0
7543 ;:*****
7544 . =60000
7545 060000 000900 LOW1: .WORD 0
7546 070000 . =70000
7547 070000 000000 HIGH1: .WORD 0
7548 000001 .END
```

SYMBOL TABLE

ABASE = 000000	BEBA = 170004	FAIL1 = 002070	SWR = 002074	TST151 = 026460
ACDW1 = 000000	BECC = 170002	FAIL2 = 002072	SWREG = 000176	TST152 = 027054
ACDW2 = 000000	BECR1 = 170006	FC = 000400	TDAR = 000001	TST153 = 027216
ACPUOP = 000000	BECR2 = 170016	FIN = 045572	TITL = 045314	TST154 = 027410
ADDW0 = 000000	BEDA = 170000	FLTPAT = 047344	TPB = 001000	TST155 = 027574
ADDW1 = 000000	BEGIN = 002416	FMHI = 000010	TPE = 000040	TST156 = 027754
ADDW10 = 000000	BIT00 = 000001	FML0 = 000004	TSTCNT = 002140	TST157 = 030102
ADDW11 = 000000	BIT01 = 000002	HIGH1 = 070000	TSTID = 000001	TST16 = 003734
ADDW12 = 000000	BIT02 = 000004	HIT = 000400	TSTIMS = 002142	TST160 = 030256
ADDW13 = 000000	BIT03 = 000010	HOD0 = 000002	TST1 = 002514	TST161 = 030422
ADDW14 = 000000	BIT04 = 000020	HPB = 004000	TST10 = 003302	TST162 = 030652
ADDW15 = 000000	BIT05 = 000040	HT = 000011	TST100 = 012646	TST163 = 031102
ADDW2 = 000000	BIT06 = 000100	INHERR = 045560	TST101 = 013072	TST164 = 031336
ADDW3 = 000000	BIT07 = 000200	KPAR0 = 172340	TST102 = 013254	TST165 = 031572
ADDW4 = 000000	BIT08 = 000400	KPAR1 = 172342	TST103 = 013572	TST166 = 032046
ADDW5 = 000000	BIT09 = 001000	KPAR2 = 172344	TST104 = 014350	TST167 = 032252
ADDW6 = 000000	BIT10 = 002000	KPAR3 = 172346	TST105 = 014562	TST17 = 004004
ADDW7 = 000000	BIT11 = 004000	KPAR4 = 172350	TST106 = 014714	TST170 = 032354
ADDW8 = 000000	BIT12 = 010000	KPAR5 = 172352	TST107 = 015052	TST171 = 032554
ADDW9 = 000000	BIT13 = 020000	KPAR6 = 172354	TST11 = 003344	TST172 = 033112
ADEVCT = 000000	BIT14 = 040000	KPAR7 = 172356	TST110 = 015236	TST173 = 033450
ADEVN = 000000	BIT15 = 100000	KPDR0 = 172300	TST111 = 015434	TST174 = 033740
ADRJMP = 002134	CA121 = 047334	KPDR1 = 172302	TST112 = 015634	TST175 = 034230
ADRSYN = 002132	CA210 = 047312	KPDR2 = 172304	TST113 = 016032	TST176 = 034606
ADRIS = 002136	CA2113 = 047324	KPDR3 = 172306	TST114 = 016232	TST177 = 035240
AENV = 000000	CCR = 177746	KPDR4 = 172310	TST115 = 016454	TST2 = 002610
AENVN = 000000	CDR = 177754	KPDR5 = 172312	TST116 = 016650	TST20 = 004054
AFATAL = 000000	CDR150 = 047330	KPDR6 = 172314	TST117 = 017242	TST200 = 035672
AM = 000010	CHR = 177752	KPDR7 = 172316	TST12 = 003424	TST201 = 036226
AMADR1 = 000000	CHRPAT = 002066	LF = 000012	TST120 = 017336	TST202 = 036562
AMADR2 = 000000	CHR157 = 047322	LOOP = 002062	TST121 = 017414	TST203 = 037116
AMADR3 = 000000	CHR50 = 047352	LOW1 = 060000	TST122 = 017522	TST204 = 037452
AMADR4 = 000000	CHR80 = 047326	LPB = 002000	TST123 = 017630	TST205 = 037574
AMAMS1 = 000000	CME = 177744	MAGPRE = 007012	TST124 = 020046	TST206 = 037716
AMAMS2 = 000000	CMPE = 100000	OFF = 001015	TST125 = 020250	TST207 = 040040
AMAMS3 = 000000	CMR = 177750	PEA = 000200	TST126 = 020466	TST21 = 004134
AMAMS4 = 000000	CMRPAT = 002064	PEHI = 000200	TST127 = 020670	TST210 = 040224
AMR210 = 047316	CMR119 = 047354	PELO = 000100	TST13 = 003520	TST211 = 040410
AMSGAD = 000000	CM1 = 100000	PRDATA = 046724	TST130 = 021136	TST212 = 040532
AMSGLG = 000000	CM1513 = 047340	PRTABL = 047246	TST131 = 021414	TST213 = 040662
AMSGTY = 000000	CM2 = 040000	PRTITL = 045752	TST132 = 021672	TST214 = 041030
AMTYP1 = 000000	CM3 = 020000	PSW = 177776	TST133 = 022110	TST215 = 041242
AMTYP2 = 000000	CNT121 = 047342	RECDAT = 047346	TST134 = 022312	TST216 = 041360
AMTYP3 = 000000	CR = 000015	RELCTH = 002370	TST135 = 022530	TST217 = 041476
AMTYP4 = 000000	CRLF = 000200	RELCTL = 002342	TST136 = 022732	TST22 = 004200
APASS = 000000	DAT = 045420	SAVR0 = 045300	TST137 = 023200	TST220 = 041760
APRIOR = 000000	DCPI = 000001	SAVR1 = 045302	TST14 = 003570	TST221 = 042126
APTC SU = 000040	DISPRE = 000174	SAVR2 = 045304	TST140 = 023456	TST222 = 042656
APTENV = 000001	EHA = 000004	SAVR3 = 045306	TST141 = 023734	TST223 = 043252
APTSIZ = 000200	ENDPAS = 044022	SAVR4 = 045310	TST142 = 024160	TST224 = 043416
APTSPO = 000100	ERROR = 104406	SAVR5 = 045312	TST143 = 024370	TST225 = 043564
ASWREG = 000000	ERRPC = 045244	SCPCND = 000004	TST144 = 024570	TST226 = 043720
ATESTN = 000000	ESA = 000020	SRO = 177572	TST145 = 025042	TST23 = 004250
AUNIT = 000000	EXDAT1 = 047336	SR3 = 172516	TST146 = 025224	TST24 = 004320
AUSWR = 000000	EXDAT3 = 047350	START = 001000	TST147 = 025544	TST25 = 004370
AVECT1 = 000000	EXDAT6 = 047332	STRTLP = 002130	TST15 = 003664	TST26 = 004440
AVECT2 = 000000	FAILAD = 047356	STRTST = 002126	TST150 = 026064	TST27 = 004510

TST3	002704	TST67	010116	\$ATY3	001620	\$ERROR	045246	\$PASS	001474
TST30	004560	TST7	003240	\$ATY4	001630	\$ETABL	001506	\$PASTM	001460
TST31	004630	TST70	010474	\$BASE	001542	\$ETEND	001612	\$QUES	002116
TST32	004700	TST71	010642	\$BELL	002112	\$FATAL	001470	\$SCPSE	002144
TST33	004750	TST72	011172	\$CDW1	001546	\$FFLG	002056	\$SWR =	160000
TST34	005020	TST73	011360	\$CDW2	001550	\$FILLC	002110	\$SWREG	001510
TST35	005070	TST74	011552	\$CHARC	044514	\$FILLS	002107	\$TESTN	001472
TST36	005140	TST75	011762	\$CPLUP	001514	\$HD =	000003	\$TIMES	044162
TST37	005210	TST76	012202	\$CRLF	002117	\$HIBTS	001452	\$TKB	002100
TST4	003000	TST77	012426	\$DBLK	045162	\$ICNT	044164	\$TKS	002076
TST40	005260	TYPDS =	104405	\$DDW0	001552	\$LF	002120	\$TN =	000227
TST41	005330	TYPE =	104401	\$DDW1	001554	\$LFLG	002055	\$TPB	002104
TST42	005400	TYPOC =	104402	\$DDW10	001576	\$MADR1	001520	\$TPFLG	002111
TST43	005450	TYPON =	104404	\$DDW11	001600	\$MADR2	001524	\$TPS	002102
TST44	005520	TYPOS =	104403	\$DDW12	001602	\$MADR3	001530	\$TRAP	045172
TST45	005570	UCB =	001000	\$DDW13	001604	\$MADR4	001534	\$TRAP2	045214
TST46	005640	UMPR00=	170200	\$DDW14	001606	\$MAIL	001466	\$TRP =	000007
TST47	005710	UMPR01=	170202	\$DDW15	001610	\$MAMS1	001516	\$TRPAD	045226
TST5	003074	UMPR02=	170204	\$DDW2	001556	\$MAMS2	001522	\$TSTM	001456
TST50	005766	UMPR03=	170206	\$DDW3	001560	\$MAMS3	001526	\$TSTNM	002060
TST51	006044	UMPR04=	170210	\$DDW4	001562	\$MAMS4	001532	\$TYPDS	044746
TST52	006122	UMPR05=	170212	\$DDW5	001564	\$MBADR	001454	\$TYPE	044166
TST53	006200	UMPR06=	170214	\$DDW6	001566	\$MFLG	002054	\$TYPEC	044400
TST54	006256	UMPR07=	170216	\$DDW7	001570	\$MSGAD	001502	\$TYPEX	044516
TST55	006334	UMPR08=	170220	\$DDW8	001572	\$MSGLG	001504	\$TYPOC	044544
TST56	006424	UMPR09=	170222	\$DDW9	001574	\$MSGTY	001466	\$TYPON	044560
TST57	006522	VCIP =	010000	\$DEVCT	001476	\$MTYP1	001517	\$TYPOS	044520
TST6	003170	VLD =	010000	\$DEVN	001544	\$MTYP2	001523	\$UNIT	001500
TST60	006572	VSU =	020000	\$DOAGN	044156	\$MTYP3	001527	\$UNITM	001462
TST61	006642	WWPD =	000100	\$DTBL	045152	\$MTYP4	001533	\$USWR	001512
TST62	006712	WWPT =	002000	\$ENDAD	044146	\$NULL	002106	\$VECT1	001536
TST63	007206	\$APTHD	001452	\$ENULL	002122	\$NWTST=	000001	\$VECT2	001540
TST64	007274	\$ATYC	001636	\$ENV	001506	\$OCNT	044742	\$OFILL	044743
TST65	007472	\$ATY1	001612	\$ENVN	001507	\$OMODE	044744	.\$X =	001452
TST66	007652								

. ABS. 070002 000  
000000 001  
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 31152 WORDS ( 122 PAGES)  
DYNAMIC MEMORY: 20346 WORDS ( 78 PAGES)  
ELAPSED TIME: 00:20:09  
CKKKAB.BIN,CKKKAB.SEQ/-SP/NL:TOC=CKKKAB.MLB/ML,CKKKAB.P11