

KK11-B

11/44 KK11B CACHE
CKKAA0

AH-F626A-MC,
COPYRIGHT 1980
FICHE 1 OF 2

JAN 1980
digital
MADE IN USA

This microfiche card contains a grid of frames. Each frame typically displays a small table or data set. The data is organized into columns and rows, with some frames containing headers or labels. The overall layout is a dense grid of information, typical of microfiche storage for large datasets or reports.

KK11-B

11/44 KK11B CACHE
CKKAA0

AH-F626A-MC

COPYRIGHT 1980
FICHE 2 OF 2

JAN 1980

digital

MADE IN USA



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

.REM %

IDENTIFICATION

PRODUCT CODE:	AC-F624A-MC
PRODUCT NAME:	CKKAAO 11/44 KK11B CACHE
DATE CREATED:	APRIL 1979
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	JOHN W. 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, BY DIGITAL EQUIPMENT CORPORATION

52
53
54
55
56
57
58

1.0 HISTORY SECTION

CKKAAO WAS RELEASED OCT 1979

60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

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 LJD FILE
 - 3.4. EXECUTION TIMES
- 4.0 ERROR INFO
 - 4.1 ERROR PRINTOUTS
 - 4.2 UNCONTROLLED ERRORS
- 5.0 HANDLERS AND COMMON ROUTINES

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
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158

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.

2.2 TEST STRUCTURE

EACH TEST IS STRUCTURED WITH THE FOLLOWING DEDICATED LOCAL
SYMBOLS:

40\$: LOCATION WHERE TEST BEGINS

159 1S: LOCATION OF THE BEGINNING OF THE LOOP ON ERROR
160 CODE LOOP
161
162 25S: LOCATION OF THE END OF THE LOOP ON ERROR CODE
163 LOOP
164
165 30S: LOCATION WHERE TEST ENDS
166

167 THESE LOCATIONS ARE USED BY THE \$SCPSET ROUTINE TO SET UP LOOP
168 ON TEST AND LOOP ON ERROR VECTORS (REFER TO \$SCPSET SECT. 9.0)
169

170 2.3 HARDWARE REQUIREMENTS

171 2.3.1 REQUIRED EQUIPMENT

- 172 1. PDP11-44 CPU
- 173 A. M7094/M7095 CPU CONTROL DATA PATH
- 174 B. M7096 MFM
- 175 C. M7098 UBI
- 176 D. M7090 CIM
- 177 2. 16K MEMORY
- 178 3. I/O TERMINAL

181 2.3.2 OPTIONAL EQUIPMENT

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

184 2.3.3 DIAGNOSTIC PREREQUISITES

185 IT IS ASSUMED THAT ALL THE ABOVE HARDWARE IS OPERATIONAL AND THAT
186 THERE RESPECTIVE DIAGNOSTICS HAVE BEEN RUN FOR VERIFICATION.
187

188 2.3.4 RELATED DOCUMENTS

- 189 1. 11/44 CACHE DESIGN SPECIFICATION
- 190 2. M7097 CACHE LOGIC SCHEMATIC
- 191 3. RMI(G5179) REGISTER DESCRIPTION
192 DATED 29 JAN 1979-PDP11 SYS. PROC. SUPPORT
- 193 4. PMK05 UNIBUS EXERCISER OPERATING AND SERVICE MANUAL
- 194 5. CFKKA0 11/34 DIAGNOSTIC
- 195
- 196
- 197
- 198
- 199
- 200
- 201
- 202
- 203
- 204
- 205

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
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263

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)

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

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
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

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
7000C.

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.

3.3 APT

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

BIT 12 \$SWR (UNIBUS EXERCISER)

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

321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368

BIT 7 \$J\$WR (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	LONGEST	ADDITIONAL
RUN TIME	TEST TIME	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

3.4 EXECUTION TIMES

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

370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403

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 ID 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	C01	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.

405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440

5.0 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).

%

555

```

.*TITLE CKKAAO 11-44 KK11B CACHE
.*COPYRIGHT (C) OCT 1979
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*

```

```

.*PROGRAM BY JOHN W. CIUKAJ
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
.*

```

000001
160000

```

$TN=1
$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

```

556
557

```

.SBTTL TRAP CATCHER

```

000000

```

.=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

```

000174 000174
000176 000000

```

.=174
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER

```

000200 000137 000200

```

.SBTTL STARTING ADDRESS(ES)
JMP @#200 ;;JUMP TO STARTING ADDRESS OF PROGRAM

```

558
565
566 000020 000020
567 000022 002134
568 000022 000340
569 000030 000030
570 000032 045252
571 000032 000340
572 000034 000340
573 000036 000340
574 000042 000042
575 000042 000000
576 000046 000046
577 000046 044154
578 000052 000052
578 000052 000000

```

.=20
$SCPSET
340
.-30
$ERROR
340
$TRAP
340
.-42
.WORD 0
.=46
.WORD $ENDAD
.52
.WORD 0

```

```

580          000200          . =200
581 000200  000137  001000  JMP START
582          001000          . =1000
583          001000          START:
584 001000  000005          RESET          ;DISABLE ALL INTERRUPTS
585 001002  012706  000500  MOV #500,SP      ;SET STACK POINTER
586
587 001006  012737  001032  000004  MOV #4$ 4        ;SFTUP FOR POSSIBLE NEX MEMORY TRAP
588 001014  012737  000340  000006  MOV #340,6
589 001022  012737  001015  177746  MOV #OFF,CCR    ;DISABLE CACHE
590 001030  000533          BR 5$           ;NO TRAP;CONTINUE
591 001032  022626          4$: CMP (SP)+,(SP)+ ;ADJUST STACK DUE TO TRAP
592 001034  012737  000006  000004  MOV #6,4        ;RESTORE TRAP VECTORS
593 001042  005037  000006
594
595 001046  104401  001054          TYPE 55$        ;:TYPE ASCIZ STRING
          001052  000416          BR 64$         ;:GET OVER THE ASCIZ
          ;:65$: .ASCIZ <CRLF>/CKKAAO 11-44 KK11B CACHE/
          64$:
596 001110  104401  001116          TYPE 67$        ;:TYPE ASCIZ STRING
          001114  000423          BR 66$         ;:GET OVER THE ASCIZ
          ;:67$: .ASCIZ <CRLF>/TRAP THRU NEX MEMORY VECTOR OCCURED/
          66$:
597 001164  104401  001172          TYPE 69$        ;:TYPE ASCIZ STRING
          001170  000424          BR 68$         ;:GET OVER THE ASCIZ
          ;:69$: .ASCIZ <CRLF>/DIAGNOSTIC ATTEMPTED TO TURN CACHE OFF/
          68$:
598 001242  104401  001250          TYPE 71$        ;:TYPE ASCIZ STRING
          001246  000423          BR 70$         ;:GET OVER THE ASCIZ
          ;:71$: .ASCIZ <CRLF>/BY ADDRESSING CACHE CONTROL REGISTER/
          70$:
          001316
599
600 001316  000000          HALT          ;HALT PROGRAM
601
602 001320  012737  000006  000004  5$: MOV #6,4        ;RESTORE VECTORS
603 001326  005037  000006          CLR 6
604 001332  005037  001466          CLR $PASS     ;CLEAR PASS COUNT
605 001336  132737  000200  001501  BITB #APTSIZE,$ENVM ;IS APT SIZING?
606 001344  001403          BEQ 1$        ;NO
607 001346  012737  001502  002064  MOV #SSWREG,SWR ;YES;USE APT SWITCH REGISTER
608 001354  005737  000042          1$: TST @#42    ;IS THIS MANUAL MODE?
609 001360  001404          BEQ 3$        ;YES TYPE ID
610 001362  023737  000042  000046  CMP @#42,@#46  ;IS THIS ACT 11 QV OR AUTO MODE?
611 001370  001423          BEQ 2$        ;YES;SKIP TITLE
612 001372  000240          3$: NOP
613 001374  000240          NOP
614 001376  104401  001404          TYPE 73$        ;:TYPE ASCIZ STRING
          001402  000416          BR 72$         ;:GET OVER THE ASCIZ
          ;:73$: .ASCIZ <CRLF>/CKKAAO 11-44 KK11B CACHE /
          72$:
          001440
615
616 001440  000137  002400          2$: JMP BEGIN    ;START TEST

```

618

.SBTTL APT PARAMETER BLOCK

```
*****  
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
*****  
:$.X=      ;;SAVE CURRENT LOCATION  
:  =24     ;;SET POWER FAIL TO POINT TO START OF PROGRAM  
: 200     ;;FOR APT START UP  
:  =44     ;;POINT TO APT INDIRECT ADDRESS PNTR.  
:$APTHDR   ;;POINT TO APT HEADER BLOCK  
:  =$.X    ;;RESET LOCATION COUNTER  
*****  
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
:INTERFACE SPEC.
```

000024 001444
000024 000024
000044 000044
000044 001444
001444 001444

```
$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)  
.SBTTL APT MAILBOX-ETABLE
```

619

```
*****  
:EVEN  
$MAIL:      ;;APT MAILBOX  
$MSGTY: .WORD AMSGTY ;;MESSAGE TYPE CODE  
$FATAL: .WORD AFATAL ;;FATAL ERROR NUMBER  
$TESTN: .WORD ATESTN ;;TEST NUMBER  
$PASS: .WORD APASS ;;PASS COUNT  
$DEVCT: .WORD ADEVCT ;;DEVICE COUNT  
$UNIT: .WORD AUNIT ;;I/O UNIT NUMBER  
$MSGAD: .WORD AMSGAD ;;MESSAGE ADDRESS  
$MSGLG: .WORD AMSGLG ;;MESSAGE LENGTH  
$ETABLE:   ;;APT ENVIRONMENT TABLE  
$ENV: .BYTE AENV ;;ENVIRONMENT BYTE  
$ENVM: .BYTE AENVM ;;ENVIRONMENT MODE BITS  
$SWREG: .WORD ASWREG ;;APT SWITCH REGISTER  
$USWR: .WORD AUSWR ;;USER SWITCHES  
$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,PDG=07,Q=10  
: *  
: * BIT 10=REAL TIME CLOCK  
: * BIT 9=FLOATING POINT PROCESSOR  
: * BIT 8=MEMORY MANAGEMENT  
$MAMS1: .BYTE AMAMS1 ;;HIGH ADDRESS,M.S. BYTE  
$MTYP1: .BYTE AMTYP1 ;;MEM. TYPE,BLK#1  
: *  
: * MEM. TYPE BYTE -- (HIGH BYTE)  
: * 900 NSEC CORE=001  
: * 300 NSEC BIPOLAR=002  
: * 500 NSEC MOS=003  
$MADR1: .WORD AMADR1 ;;HIGH ADDRESS,BLK#1  
: *  
: * MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE  
$MAMS2: .BYTE AMAMS2 ;;HIGH ADDRESS,M.S. BYTE  
$MTYP2: .BYTE AMTYP2 ;;MEM. TYPE,BLK#2
```

001460 000000
001460 000000
001462 000000
001464 000000
001466 000000
001470 000000
001472 000000
001474 000000
001476 000000
001500
001500 000
001501 000
001502 000000
001504 000000
001506 000000

001510 000
001511 000

001512 000000

001514 000
001515 000

001516	000000	\$MADR2:	.WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
001520	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
001521	000	\$MTYP3:	.BYTE	AMTYP3	::MEM.TYPE,BLK#3
001522	000000	\$MADR3:	.WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
001524	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
001525	000	\$MTYP4:	.BYTE	AMTYP4	::MEM.TYPE,BLK#4
001526	000000	\$MADR4:	.WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
001530	000000	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
001532	000000	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
001534	000000	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
001536	000000	\$DEVN:	.WORD	ADEVN	::DEVICE MAP
001540	000000	\$CDW1:	.WORD	ACDW1	::CONTROLLER DESCRIPTION WORD#1
001542	000000	\$CDW2:	.WORD	ACDW2	::CONTROLLER DESCRIPTION WORD#2
001544	000000	\$DDW0:	.WORD	ADDW0	::DEVICE DESCRIPTOR WORD#0
001546	000000	\$DDW1:	.WORD	ADDW1	::DEVICE DESCRIPTOR WORD#1
001550	000000	\$DDW2:	.WORD	ADDW2	::DEVICE DESCRIPTOR WORD#2
001552	000000	\$DDW3:	.WORD	ADDW3	::DEVICE DESCRIPTOR WORD#3
001554	000000	\$DDW4:	.WORD	ADDW4	::DEVICE DESCRIPTOR WORD#4
001556	000000	\$DDW5:	.WORD	ADDW5	::DEVICE DESCRIPTOR WORD#5
001560	000000	\$DDW6:	.WORD	ADDW6	::DEVICE DESCRIPTOR WORD#6
001562	000000	\$DDW7:	.WORD	ADDW7	::DEVICE DESCRIPTOR WORD#7
001564	000000	\$DDW8:	.WORD	ADDW8	::DEVICE DESCRIPTOR WORD#8
001566	000000	\$DDW9:	.WORD	ADDW9	::DEVICE DESCRIPTOR WORD#9
001570	000000	\$DDW10:	.WORD	ADDW10	::DEVICE DESCRIPTOR WORD#10
001572	000000	\$DDW11:	.WORD	ADDW11	::DEVICE DESCRIPTOR WORD#11
001574	000000	\$DDW12:	.WORD	ADDW12	::DEVICE DESCRIPTOR WORD#12
001576	000000	\$DDW13:	.WORD	ADDW13	::DEVICE DESCRIPTOR WORD#13
001600	000000	\$DDW14:	.WORD	ADDW14	::DEVICE DFSCRIPTOR WORD#14
001602	000000	\$DDW15:	.WORD	ADDW15	::DEVICE DESCRIPTOR WORD#15

001604

\$ETEND:

620

.\$BTTL APT COMMUNICATIONS ROUTINE

```

*****
001604 112737 000001 002050 $ATY1: MOV  #1,$FFLG ;;TO REPORT FATAL ERROR
001612 112737 000001 002046 $ATY3: MOV  #1,$MFLG ;;TO TYPE A MESSAGE
001620 000403          BR      $ATYC
001622 112737 000001 002050 $ATY4: MOV  #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
001630          $ATYC:
001630          MOV  R0,-(SP) ;;PUSH R0 ON STACK
001632          MOV  R1,-(SP) ;;PUSH R1 ON STACK
001634 105737 002046          TSTB  $MFLG ;;SHOULD TYPE A MESSAGE?
001640 001450          BEQ   $S ;;IF NOT: BR
001642 122737 000001 001500 CMPB  #APTENV,$ENV ;;OPERATING UNDER APT?
001650 001031          BNE   $S ;;IF NOT: BR
001652 132737 000100 001501 BITB  #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
001660 001425          BEQ   $S ;;IF NOT: BR
001662 017600 000004          MOV  @4(SP),R0 ;;GET MESSAGE ADDR.
001666 062766 000002 000004 ADD  #2,4(SP) ;;BUMP RETURN ADDR.
001674 005737 001460          $S: TST  $MSGTYPE ;;SEE IF DONE W/ LAST XMIS$ION?
001700 001375          BNE   $S ;;IF NOT: WAIT
001702 010037 001474          MOV  R0,$MSGAD ;;PUT ADDR IN MAILBOX
001706 105720          $S: TSTB (R0)+ ;;FIND END OF MESSAGE
001710 001376          BNE   $S
001712 163700 001474          SUB  $MSGAD,R0 ;;SUB START OF MESSAGE

```



```
001716 006200 ASR R0 ;;GET MESSAGE LNGTH IN WORDS
001720 010037 001476 MOV R0,$MSGJLGT ;;PUT LENGTH IN MAILBOX
001724 012737 000004 001460 MOV #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
001732 000413 BR 5$
001734 017637 000004 001760 3$: MOV @4(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
001742 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDRESS
001750 013746 177776 MOV 177776,-(SP) ;;PUSH 177776 ON STACK
001754 004737 044174 JSR PC,$TYPE ;;CALL TYPE MACRO
001760 000000 4$: .WORD 0
001762 5$:
001762 105737 002050 10$: TSTB $FFLG ;;SHOULD REPORT FATAL ERROR?
001766 001416 BEQ 12$ ;;IF NOT: BR
001770 005737 001500 TST $ENV ;;RUNNING UNDER APT?
001774 001413 BEQ 12$ ;;IF NOT: BR
001776 005737 001460 11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
002002 001375 BNE 11$ ;;IF NOT: WAIT
002004 017637 000004 001462 MOV @4(SP),$FATAL ;;GET ERROR #
002012 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
002020 005237 001460 INC $MSGTYPE ;;TELL APT TO TAKE ERROR
002024 105037 002050 12$: CLRB $FFLG ;;CLEAR FATAL FLAG
002030 105037 002047 CLRB $LFLG ;;CLEAR LOG FLAG
002034 105037 002046 CLRB $MFLG ;;CLEAR MESSAGE FLAG
002040 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
002042 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
002044 000207 RTS PC ;;RETURN
002046 000 SMFLG: .BYTE 0 ;;MESSG. FLAG
002047 000 $LFLG: .BYTE 0 ;;LOG FLAG
002050 000 $FFLG: .BYTE 0 ;;FATAL FLAG
.EVEN
```

```
000200
000001
000100
000040
APTSIZE=200
APTENV=001
APTSPOOL=100
APTCSUP=040
```

```
621
622
623 :*****
624 : USER LABELS
625 :*****
```

```
626 002052 000000 LOOP: .WORD 0
627 002054 000000 CMRPAT: .WORD 0
628 002056 000000 CHRPAT: .WORD 0
629 002060 000000 FAIL1: .WORD 0
630 002062 000000 FAIL2: .WORD 0
631
632
633 002064 177570 SWR: .WORD 177570
634 002066 177560 $TKS: 177560
635 002070 177562 $TKB: 177562
636 002072 177564 $TPS: 177564
637 002074 177566 $TPB: 177566
638 002076 000 $NULL: .BYTE 0
639 002077 002 $FILLS: .BYTE 2
640 002100 012 $FILLC: .BYTE 12
641 002101 000 $TPFLG: .BYTE 0
642 002102 207 377 377 $BELL: .ASCII <207><377><377>
643 002106 077 $QUES: .ASCII /?/
```

644	002107	015			\$CRLF:	.ASCII	<15>
645	002110	012	000		\$LF:	.ASCII	<12>
646	002112	377	377	000	\$ENULL:	.BYTE	-1,-1,0
647						.EVEN	

649
650
651
652 177744
653 177746
654 177750
655 177752
656 177754
657 177776
658 000000
659 000001
660 000002
661 000003
662 000004
663 000005
664 000006
665 000007
666 000001
667 000002
668 000004
669 000010
670 000020
671 000040
672 000100
673 000200
674 000400
675 001000
676 002000
677 004000
678 010000
679 020000
680 040000
681 100000
682
683
684 172300
685 172302
686 172304
687 172306
688 172310
689 172312
690 172314
691 172316
692 172340
693 172342
694 172344
695 172346
696 172350
697 172352
698 172354
699 172356
700 177572
701 172516
702 170200
703 170202
704 170204
705 170206

.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
: GENERAL REGISTERS
R0 = %0
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
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

706	170210	UMPR04=170210
707	170212	UMPR05=170212
708	170214	UMPR06=170214
709	170216	UMPR07=170216
710	170220	UMPR08=170220
711	170222	UMPR09=170222
712	170002	BECC = 170002
713	170004	BEBA = 170004
714	170000	BEDA = 170000
715	170006	BECR1 = 170006
716	170016	BECR2 = 170016
717		
718	000200	APTSIZE=200
719	000001	APTENV=001
720	000100	APTSPOOL=100
721	000040	APTSUP=040
722		
723		:CCR REGISTER
724	000001	DCPI=1
725	000004	FMLO=4
726	000010	FMHI=10
727	000100	WWPD=100
728	000200	PEA=200
729	000400	FC=400
730	001000	UCB=1000
731	002000	WWPT=2000
732	010000	VCIP=10000
733	020000	VSIU=20000
734		
735		:CMR REGISTER
736	000001	TDAR=1
737	000002	HODO=2
738	000004	EHA=4
739	000010	AM=10
740	000020	ESA=20
741	000400	HIT=400
742	001000	TPB=1000
743	002000	LPB=2000
744	004000	HPB=4000
745	010000	VLD=10000
746	020000	CM3=20000
747	040000	CM2=40000
748	100000	CM1=100000
749		
750		:CME REGISTER
751	000040	TPE=40
752	000100	PELO=100
753	000200	PEHI=200
754	100000	CMPE=100000
755		
756	000001	TSTID=1
757	000004	SCPCND=4
758	001015	OFF=1015
759		

761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817

002116 000000
002120 000000
002122 000000
002124 000000
002126 000000
002130 000000
002132 000000

002134
002134 012737 000340 177776
002142 022737 000001 001466
002150 001434
002152 032777 040000 177704
002160 001413
002162 013702 001464
002166 005302
002170 120277 177670
002174 001005
002176 022626
002200 005337 001464
002204 000177 177706

002210 005737 001466
002214 001412
002216 032777 004000 177640
002224 001006
002226 005237 002130
002232 023737 002132 002130
002240 001356

002242 005037 002130
002246 011601
002250 012137 002116
002254 012137 002120
002260 012137 002122
002264 012137 002124

013737 002124 002126
062737 000002 002126
012777 000240 177612
012777 000240 177606
010116
000002

```

:*****
:          SETUP TEST CONDITIONS:
:          1. TEST ITERATIONS
:          2. LOAD TEST VECTORS FOR LOOP ON TEST,
:             LOOP ON ERROR
:*****

```

```

STRST: .WORD 0
STRSLP: .WORD 0
ADRSYNC: .WORD 0
ADRJMP: .WORD 0
ADR1$: .WORD 0
TSTCNT: .WORD 0
TSTIMS: .WORD 1.

```

```

$SCPSET:
MOV #340,PSW           ;CPU HI PRIORITY
CMP #1,$TESTN         ;IS THIS TEST 1?
BEQ 3$                ;YES,DO NOT CONSIDER LOOP ON TEST
BIT #BIT14,@SWR       ;LOOP ON TEST?
BEQ 4$                ;NO
MOV $TESTN,R2         ;GET PRESENT TEST NUMBER
DEC R2                ;GET LAST TEST NUMBER
CMPB R2,@SWR          ;IS THIS THE TEST?
BNE 4$                ;NO
5$: CMP (SP)+,(SP)+    ;YES;PREPARE FOR LOOP ON TEST
DEC $TESTN
JMP @STRST            ;GO LOOP ON TEST

4$: TST $PASS          ;FIRST PASS?
BEQ 3$                ;YES;INHIBIT TEST ITERATIONS
BIT #BIT11,@SWR       ;INHIBIT ITERATIONS?
BNE 3$                ;YES
INC TSTCNT            ;INCREMENT TEST ITERATION COUNTER
CMP TSTIMS,TSTCNT     ;ITERATIONS COMPLETE?
BNE 5$                ;NO CONTINUE WITH TEST

3$: CLR TSTCNT
MOV (SP),R1           ;GET ADDRESS OF FIRST ARGUMENT
MOV (R1)+,STRST       ;LOCATION OF START OF TEST
MOV (R1)+,STRSLP      ;LOCATION OF START OF SCOPE LOOP
MOV (R1)+,ADRSYNC     ;ADDRESS LOADED INTO AMR FOR SCOPE SYNC
MOV (R1)+,ADRJMP      ;ADDRESS OF END OF SCOPE LOOP AND
                       ;WHERE 'JMP' IS WRITTEN
MOV ADRJMP,ADR1$      ;
ADD #2,ADR1$          ;LOCATION WHERE '1$' IS WRITTEN
MOV #240,@ADRJMP      ;INITIALIZE SCOPE LOCATIONS
MOV #240,@ADR1$
MOV R1,(SP)           ;SETUP STACK FOR RETURN
RTI

```

818
819
820
821
822
823
824 002324 012701 060000
825 002330 012402
826 002332 012421
827 002334 020402
828 002336 001375
829 002340 013721 002350
830 002344 000137 060000
831 002350 000204
832
833
834 002352 012701 070000
835 002356 012402
836 002360 012421
837 002362 020402
838 002364 001375
839 002366 013721 002376
840 002372 000137 070000
841 002376 000204
842
843
844
845

.....
:.....
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

847 002400 012706 000500
848 002404 000005
849 002406 012737 000340 177776
850 002414 012737 001015 177746
851 002422 005037 001464
852 002426 012737 000002 000000
853 002434 005037 000002
854 002440 012737 000006 000004
855 002446 005037 000006
856 002452 012737 000116 000114
857 002460 005037 000116
858 002464 005037 002052
859 002470 005337 002052
860 002474 001375
861
862

BEGIN:

```
MOV #500,SP           ;SET UP STACK
RESET
MOV #340,PSW         ;CPU HI PRIORITY
MOV #OFF,CCR        ;DISABLE CACHE
CLR $TESTN          ;RESET TEST ID COUNTER
MOV #2,@#0          ;INITIALIZE A FEW VECTORS
CLR @#2
MOV #6,@#4
CLR @#6
MOV #116,@#114
CLR @#116
CLR LOOP            ;SOFTWARE DELAY
DEC LOOP
BNE 'S
```

864
865
866
867
868
869
870 002476 005237 001464

.SBTTL *
.SBTTL *
CACHE REGISTER RESPONSE TESTS
.SBTTL *
.SBTTL *

T1: INC \$TESTN ;UPDATE TEST ID
:*****

871
872
873
874
875
876 002502 000004

.SBTTL
.SBTTL TEST 1
.SBTTL
:*****
.SBTTL ATTEMPT READ INTO CME TO TEST ADDRESS SELECT LOGIC
.SBTTL IF TIME OUT OCCURES THEN LOGIC IN FAULT
.SBTTL
:*****

002504 002514
002506 002514
002510 000000
002512 002540
002514

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

877 002514 012737 002546 000004
878 002522 012737 000340 000006
879 002530 005737 177744
880 002534 000240
881 002536 000240
882

40\$:
1\$: MOV #2\$,4 ;SETUP TRAP VECTOR
MOV #340,6
*ST CME ;READ PARITY REGISTER
NOP
NOP

002540 000240
002542 000240

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

883 002544 000411
884 002546 022626
885 002550 012737 000006 000004
886 002556 005037 000006
887

BR 10\$;NO FAULT;GO TO NEXT TEST
2\$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
MOV #6,4 ;RESTORE TRAP VECTOR
CLR 6

002562 104000
002564 002562

ERROR ;ERROR
;-2

888
889
890
891
892 002566 000000
893 002570 000240
894
895

;CACHE REGISTER RESPONSE TESTS
;READING PARITY FAULT REGISTER
;CAUSED TIMEOUT

896 002572 005237 001464

T2: INC \$TESTN ;UPDATE TEST ID
:*****

897
898
899
900

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


```

901          ;;*****
902 002576 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          002600 002610          40$          ;TEST START LOCATION
          002602 002610          1$          ;LOOP ON ERROR START LOCATION
          002604 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          002606 002634          25$          ;LOOP ON ERROR END LOCATION
          002610          40$:
903 002610 012737 002642 000004          1$: MOV #2$,4          ;SETUP TRAP VECTOR
904 002616 012737 000340 000006          MOV #340,6
905 002624 005737 177746          TST CCR          ;READ CACHE CONTROL REGISTER
906 002630 000240          NOP
907 002632 000240          NOP
908          002634 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          002636 000240          NOP          ;FOR LOOP ON ERROR
909 002640 000411          BR 10$          ;NO FAULT;GO TO NEXT TEST
910 002642 022626          2$: CMP (SP)+,(SP)+          ;READJUST STACK DUE TO INTERRUPT
911 002644 012737 000006 000004          MOV #6,4          ;RESTORE TRAP VECTOR
912 002652 005037 000006          CLR 6
913          002656 104000          ERROR          ;ERROR
          ;-----
          002660 002656          -2
914          ;CACHE REGISTER RESPONSE TESTS
915          ;READING CACHE CONTROL REGISTER
916          ;CAUSED TIMEOUT
917
918 002662 000000          0
919 002664 000240          10$: NOP          ;END OF TEST
920
921
922 002666 005237 001464          T3: INC $TESTN          ;UPDATE TEST ID
          ;;*****
          .SBTTL
          .SBTTL TEST 3
          .SBTTL
          ;;*****
923          .SBTTL ATTEMPT READ INTO CMR TO CHECK ADDRESS SELECT LOGIC
924          .SBTTL IF TIME OCCURES THEN LOGIC IN FAULT
925          .SBTTL
926          .SBTTL
927          ;;*****
928 002672 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          002674 002704          40$          ;TEST START LOCATION
          002676 002704          1$          ;LOOP ON ERROR START LOCATION
          002700 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          002702 002730          25$          ;LOOP ON ERROR END LOCATION
          002704          40$:
929 002704 012737 002736 000004          1$: MOV #2$,4          ;SETUP TRAP VECTOR
930 002712 012737 000340 000006          MOV #340,6
931 002720 005737 177750          TST CMR          ;READ MAINTENANCE REGISTER
932 002724 000240          NOP
933 002726 000240          NOP
934          002730 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE

```

```

002732 000240          NOP          ;FOR LOOP ON ERROR
935 002734 000411          BR 10$          ;NO FAULT;GO TO NEXT TEST
936 002736 022626          2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
937 002740 012737 000006 000004    MOV #6,4      ;RESTORE TRAP VECTOR
938 002746 005037 000006          CLR 6
939
002752 104000          ERROR          ;ERROR
                                ;-----
002754 002752          .-2
940                                ;CACHE REGISTER RESPONSE TESTS
941                                ;READING MAINTENANCE REGISTER
942                                ;CAUSED TIMEOUT
943
944 002756 000000          0
945 002760 000240          10$: NOP          ;END OF TEST
946
947
948 002762 005237 001464    T4:          INC $TESTN          ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 4
                                .SBTTL
949                                ;*****
950                                .SBTTL ATTEMPT READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
951                                .SBTTL IF TIME OUT OCCURES THEN LOGIC IN FAULT
952                                .SBTTL
953                                ;*****
954 002766 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                002770 003000          40$          ;TEST START LOCATION
                                002772 003000          1$          ;LOOP ON ERROR START LOCATION
                                002774 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                002776 003024          25$          ;LOOP ON ERROR END LOCATION
                                003000          40$:
955 003000 012737 003032 000004    1$: MOV #2$,4      ;SETUP TRAP VECTOR
956 003006 012737 000340 000006    MOV #340,6
957 003014 005737 177752          TST CHR      ;READ HIT REGISTER
958 003020 000240          NOP
959 003022 000240          NOP
960
003024 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
003026 000240          NOP          ;FOR LOOP ON ERROR
961 003030 000411          BR 10$          ;NO FAULT;GO TO NEXT TEST
962 003032 022626          2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
963 003034 012737 000006 000004    MOV #6,4      ;RESTORE TRAP VECTOR
964 003042 005037 000006          CLR 6
965
003046 104000          ERROR          ;ERROR
                                ;-----
003050 003046          .-2
966                                ;CACHE REGISTER RESPONSE TESTS
967                                ;READING HIT REGISTER
968                                ;CAUSED TIMEOUT
969
970 003052 000000          0

```

```

971 003054 000240          10$:  NOP          ;END OF TEST
972
973
974 003056 005237 001464  T5:          INC $TESTN          ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 5
.SBTTL
975
976 .SBTTL ATTEMPT READ INTO CDR TO CHECK ADDRESS SELECT LOGIC
977 .SBTTL IF TIMEOUT OCCURS THEN LOGIC IN FAULT
978 .SBTTL
979 ;*****
980 003062 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
003064 003074          40$          ;LOOP ON ERROR START LOCATION
003066 003074          1$          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
003070 000000          0          ;LOOP ON ERROR END LOCATION
003072 003120          25$
003074          40$:
981 003074 012737 003126 000004  1$:  MOV #2$,4          ;SETUP TRAP VECTOR
982 003102 012737 000340 000006  MOV #340,6
983 003110 005737 1777  TST CDR          ;READ DATA REGISTER
984 003114 000240          NOP
985 003116 000240          NOP
986
003120 000240          25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
003122 000240          NOP          ;FOR LOOP ON ERROR
987 003124 000411          BR 10$          ;NO FAULT;GO TO NEXT TEST
988 003126 022626          2$:  CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
989 003130 012737 000006 000004  MOV #6,4          ;RESTORE TRAP VECTOR
990 003136 005037 000006
991
003142 104000          ERROR          ;ERROR
;-----
003144 003142          -2          ;CACHE REGISTER RESPONSE TESTS
;READING DATA REGISTER
;CAUSED TIMEOUT
992
993
994
995
996 003146 000000          0
997 003150 000240          10$:  NOP          ;END OF TEST
998
999
1000 003152 005237 001464  T6:          INC $TESTN          ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 6
.SBTTL
1001
1002 .SBTTL TESTING ADDRESS SELECTION LOGIC BY WRITING ONE INTO UNUSED
1003 .SBTTL CME REGISTER BIT00 THEN READ CONTENTS OF REGISTER BACK
1004 .SBTTL LOOKING TO SEE IF BIT00 STILL READ, AS 0.
1005 .SBTTL IF BIT00 IS SET IT IS POSSIBLE WE ARE ADDRESSING THE WRONG
1006 .SBTTL REGISTER
1007 ;*****

```

```

1008 003156 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          003160 003170          40$          ;ERROR/LOOP ON TEST
          003162 003170          1$          ;TEST START LOCATION
          003164 000000          0          ;LOOP ON ERROR START LOCATION
          003166 003176          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          003170          40$:          ;LOOP ON ERROR END LOCATION
1009 003170 012737 000001 177744 1$:      MOV #1,CME          ;WRITE 1 INTO BIT00
1010          003176 000240          25$:      NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          003200 000240          NOP          ;FOR LOOP ON ERROR

1011 003202 032737 000001 177744          BIT #1,CME          ;CHECK FOR 0
1012 003210 001403          BEQ 10$         ;PASS;NEXT TEST
1013          003212 104000          ERROR          ;ERROR
          003214 003212          .-2          ;-----

1014          ;CACHE REGISTER RESPONSE TESTS
1015          ;UNUSED CME BIT00 READ AS 1
1016          ;POSSIBLE REG. ADDRESS ERROR
1017
1018 003216 000000          0
1019 003220 000240          10$:      NOP          ;END OF TEST
1020
1021
1022 003222 005237 001464          17.          INC $TESTN          ;UPDATE TEST ID
          ;*****
          ;SBTTL
          ;SBTTL TEST 7
          ;SBTTL
          ;*****
          ;SBTTL ASSURING BIT00(DCPI) READS AS A 1 AND TESTING ADDRESS
          ;SBTTL SELECT LOGIC BY WRITING 1 INTO BIT00 OF CCR AND THEN
          ;SBTTL READING A 1. IF BIT00 READS AS 0 POSSIBLE ADDRESSING
          ;SBTTL WRONG REGISTER OR CCR REGISTER/DATA PATH ARE BAD.
          ;SBTTL
          ;*****
1030 003226 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          003230 003240          40$          ;ERROR/LOOP ON TEST
          003232 003240          1$          ;TEST START LOCATION
          003234 000000          0          ;LOOP ON ERROR START LOCATION
          003236 003246          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          003240          40$:          ;LOOP ON ERROR END LOCATION
1031 003240 032737 000001 177746 1$:      BIT #1,CCR          ;CHECK BIT00 FOR 1
1032          003246 000240          25$:      NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          003250 000240          NOP          ;FOR LOOP ON ERROR

1033 003252 001003          BNE 10$         ;PASS;NXT TEST
1034          003254 104000          ERROR          ;ERROR
          003256 003254          .-2          ;-----

1035          ;CACHE REGISTER RESPONSE TESTS
1036          ;WROTE 1 INTO BIT00 CCR; READ

```

```

1037
1038 003260 000000
1039 003262 000240          10$:      0      ;END OF TEST
1040
1041
1042          .SBTTL *
1043          .SBTTL
1044          .SBTTL *          CACHE CONTROL REGISTER DATA TEST (CCR)
1045          .SBTTL
1046          .SBTTL *
1047
1048
1049 003264 005237 001464    T10:          INC $TESTN          ;UPDATE TEST ID
          :*****
          .SBTTL
          .SBTTL TEST 10
          .SBTTL
1050          :*****
1051          .SBTTL VERIFY THAT CCR BIT12(VCIP) READS AS A 0, SINCE A CLEARING
1052          .SBTTL OF VALID STORE SHOULD NOT BE HAPPENING AT THIS TIME
1053          .SBTTL
1054          :*****
1055 003270 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          :ERROR/LOOP ON TEST
          003272 003302          40$          ;TEST START LOCATION
          003274 003302          1$          ;LOOP ON ERROR START LOCATION
          003276 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          003300 003310          25$          ;LOOP ON ERROR END LOCATION
          003302
1056 003302 032737 010000 177746    40$:          ;CHECK FOR 0
          1$:          BIT #BIT12,CCR
          1057          25$:          ;INSTRUCTION 'JMP 1$' PLACED HERE
          003310 000240          NOP          ;FOR LOOP ON ERROR
          003312 000240
1058 003314 001403          BEQ 10$          ;PASS
1059
          003316 104000          ERROR          ;ERROR
          003320 003316          .-2          ;-----
1060          ;CCR DATA TEST
1061          ;READ 1 FROM CCR BIT12. A CLEARING OF
1062          ;VALID STORE AT THIS TIME SHOULD NOT
1063          ;BE INDICATED
1064 003322 000000          0
1065 003324 000240          10$:      0      ;NXT TEST
1066
1067
1068
1069 003326 005237 001464    T11:          INC $TESTN          ;UPDATE TEST ID
          :*****
          .SBTTL
          .SBTTL TEST 11
          .SBTTL
1070          :*****
1071          .SBTTL WRITE ZERO INTO CCR BIT00 THEN READ CCR
1072          .SBTTL IF CCR IS READ AS ONE THEN CACHE CCR REGISTER MAY BE BAD
1073          .SBTTL OR CACHE REGISTER DATA PATH COULD BE IN ERROR

```

```

1074
1075 003332 000004          ;:*****
                                SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                               ;ERROR/LOOP ON TEST
                                40$          ;TEST START LOCATION
                                1$          ;LOOP ON ERROR START LOCATION
                                0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$        ;LOOP ON ERROR END LOCATION
                                40$:        MOV #OFF+BIT08,CCR          ;DISABLE AND FLUSH CACHE
                                1$:        BIC #BIT00,CCR             ;WRITE 0
                                25$:        MOV CCR,R0                ;SAVE CCR CONTENTS
                                25$:        NOP                        ;INSTRUCTION 'JMP '$' PLACED HERE
                                25$:        NOP                        ;FOR LOOP ON ERROR
1076 003344 012737 001415 177746
1077 003352 042737 000001 177746
1078 003360 013700 177746
1079
                                25$:        NOP
                                25$:        NOP
                                1080 003370 032700 000001          BIT #BIT00,R0          ;CHECK FOR 0
                                1081 003374 001403          BEQ 10$              ;PASS; NXT TEST
                                1082
                                003376 104000          ERROR                ;ERROR
                                003400 003376          .-2                    ;-----
                                1083
                                1084
                                1085
                                1086 003402 000000
                                1087 003404 000240          10$:        NOP          ;END OF TEST
                                1088
                                1089
                                1090 003406 005237 001464          T12:        INC $TESTN          ;UPDATE TEST ID
                                ;:*****
                                .SBTTL
                                .SBTTL TEST 12
                                .SBTTL
                                ;:*****
                                .SBTTL WRITE ZERO INTO CCR BIT02(FMLO) THEN READ CCR
                                .SBTTL IF BIT02 IS READ AS ONE THEN CCR REGISTER MAY BE BAD
                                .SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
                                ;:*****
1091
1092
1093
1094
1095
1096 003412 000004          ;:*****
                                SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                               ;ERROR/LOOP ON TEST
                                40$          ;TEST START LOCATION
                                1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
                                0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
                                40$:        MOV #OFF,CCR             ;DISABLE CACHE
                                40$:        JSR R4,RELCTH            ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                10$:        10$+2                    ;ADDRESS OF START OF NEXT TEST
                                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                ;ARE RELOCATED TO HI CACHE SPACE
1097 003440 042737 000004 177746
1098
                                1$:        BIC #BIT02,CCR             ;WRITE 0
                                25$:        NOP                        ;INSTRUCTION 'JMP 1$' PLACED HERE
                                25$:        NOP                        ;FOR LOOP ON ERROR
1099 003452 013701 177746          MOV CCR,R1          ;SAVE CCR CONTENTS
    
```

```

1100 003456 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
1101 003464 032701 000004              BIT #BIT02,R1    ;CHECK FOR 0
1102 003470 001403                      BEQ 10$          ;PASS; NXT TEST
1103                                ERROR                ;ERROR
                                .-2                ;-----
003472 104000
003474 003472                          .-2
1104                                ;CCR DATA TEST
1105                                ;WROTE 0 INTO CCR BIT02; READ 1
1106
1107 003476 000000                          0
1108 003500 000240                          10$: NOP          ;END OF TEST
1109
1110
1111 003502 005237 001464      T13:          INC $TESTN      ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 13
                                .SBTTL
1112                                ;*****
1113                                .SBTTL WRITE ONE INTO CCR BIT02(FMLO) AND ASSURE THAT IT READS 1.
1114                                .SBTTL IF READS BACK AS 0 THEN CCR MAY BE BAD OR CACE DATA PATH
1115                                .SBTTL IS AT FAULT
1116                                ;*****
1117 003506 000004                      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                40$                ;ERROR/LOOP ON TEST
                                1$                ;TEST START LOCATION
                                0                 ;LOOP ON ERROR START LOCATION
                                25$               ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$               ;LOOP ON ERROR END LOCATION
003510 003520                          40$
003512 003520                          1$
003514 000000                          0
003516 003526                          25$
003520                          40$:
1118 003520 052737 000004 177746      1$: BIS #FMLO,CCR      ;
1119                                25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
                                003526 000240      NOP          ;FOR LOOP ON ERROR
                                003530 000240
1120 003532 032737 000004 1777-6      BIT #BIT02,CCR    ;CHECK FOR 1
1121 003540 001003                      BNE 10$          ;PASS
1122                                ERROR                ;ERROR
                                .-2                ;-----
003542 104000
003544 003542                          .-2
1123                                ;CCR DATA TEST
1124                                ;WROTE 1 INTO CCR BIT02; READ 0
1125
1126 003546 000000                          0
1127 003550 000240                          10$: NOP          ;END OF TEST
1128
1129
1130
1131
1132 003552 005237 001464      T14:          INC $TESTN      ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 14
                                .SBTTL
1133                                ;*****
    
```

```
1134 .SBTTL WRITE ZERO INTO CCR BIT03(FMHI) THEN READ CCR
1135 .SBTTL IF BIT03 READ BACK AS ONE THEN CCR REGISTER BIT MAY BE BAD
1136 .SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
1137 ;*****
1138 003556 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                40$          ;TEST START LOCATION
                                1$-40$+60000-14 ;LOOP ON ERROR START LOCATION
                                0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$-40$+60000-14 ;LOOP ON ERROR END LOCATION
003560 003570          40$          ;DISABLE CACHE
003562 060000          1$-40$+60000-14 ;LOCATE TEST CODE TO LOW CACHE SPACE
003564 000000          0           ;ADDRESS OF START OF NEXT TEST
003566 060012          25$-40$+60000-14
003570 012737 001015 177746 40$: MOV #OFF,CCR
003576 004437 002324      JSR R4,RELCTL
003602 003646          10$+2

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

1139 003604 042737 000010 177746 1$: BIC #BIT03,CCR ;WRITE 0
1140 003612 013700 177746      MOV CCR,R0 ;SAVE CONTENTS OF CCR
1141
003616 000240          25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
003620 000240          NOP ;FOR LOOP ON ERROR

1142 003622 012737 001015 177746 MOV #OFF,CCR ;:DISABLE CACHE
1143 003630 0377'0 000010      BIT #BIT03,R0 ;CHECK FOR 0
1144 003634 001403          BEQ 10$ ;PASS
1145
003636 104000          ERROR ;ERROR
                                ;-----
003640 003636          .-2

1146 ;CCR DATA TEST
1147 ;WROTE 0 INTO CCR BIT03; READ 1
1148
1149 003642 000000          0
1150 003644 000240          10$: NOP ;END OF TEST
1151
1152
1153
1154 003646 005237 001464 T15: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 15
.SBTTL
;*****
1155 .SBTTL WRITE 1 INTO CCR BIT03(FMHI) AND ASSURE IT READS 1.
1156 .SBTTL IF CCR BIT03 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
1157 .SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
1158 ;*****
1159 ;*****
1160 003652 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                40$          ;TEST START LOCATION
                                1$          ;LOOP ON ERROR START LOCATION
                                0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$          ;LOOP ON ERROR END LOCATION
003654 003664          40$
003656 003664          1$
003660 000000          0
003662 003672          25$
003664          40$:
1161 003664 052737 000010 177746 1$: BIS #FMHI,CCR
1162
```



```

003672 000240          25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
003674 000240          NOP          ;FOR LOOP ON ERROR

1163 003676 032737 000010 177746  BIT #BIT03,CCR  ;CHECK FOR 1
1164 003704 001003          BNE 10$        ;PASS
1165                                ERROR          ;ERROR
                                .-2          ;-----

1166                                .-2          ;CCR DATA TEST
1167                                ;WROTE 1 INTO CCR BIT03; READ 0
1168
1169 003712 000000          0
1170 003714 000240          10$:  NOP          ;END OF TEST
1171
1172
1173 003716 005237 001464 116:  INC $TESTN      ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 16
                                .SBTTL
1174                                ;*****
1175                                .SBTTL WRITE 0 INTO CCR BIT06(WWPD) THEN READ CCR
1176                                .SBTTL IF BIT06 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
1177                                .SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
1178                                ;*****
1179 003722 000004          SCPCND        ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                40$          ;TEST START LOCATION
                                1$          ;LOOP ON ERROR START LOCATION
                                0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$        ;LOOP ON ERROR END LOCATION

1180 003734 042737 000100 177746  40$:  BIC #BIT06,CCR ;WRITE 0
1181                                1$:
                                25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP          ;FOR LOOP ON ERROR

1182 003746 032737 000100 177746  BIT #BIT06,CCR  ;CHECK FOR 0
1183 003754 001403          BEQ 10$        ;PASS
1184                                ERROR          ;ERROR
                                .-2          ;-----

1185                                .-2          ;CCR DATA TEST
1186                                ;WROTE 0 INTO CCR BIT06; READ 1
1187
1188 003762 000000          0
1189 003764 000240          10$:  NOP          ;END OF TEST
1190
1191
1192
1193
1194 003766 005237 001464 117:  INC $TESTN      ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 17
    
```

```

1195 .SBTTL
1196 .:*****
1197 .SBTTL WRITE ZERO INTO CCR BIT07(PEA) THEN READ CCR
1198 .SBTTL IF CCR BIT07 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
1199 .SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
1200 .:*****
1200 003772 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          40$              ;ERROR/LOOP ON TEST
          1$              ;TEST START LOCATION
          0                ;LOOP ON ERROR START LOCATION
          25$             ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          40$:            ;LOOP ON ERROR END LOCATION
          1$:            BIC #BIT07,CCR          ;WRITE 0
1201 004004 042737 000200 177746
1202          25$:            NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
          004012 000240          NOP                ;FOR LOOP ON ERROR
          004014 000240
1203 004016 032737 000200 177746          BIT #BIT07,CCR          ;CHECK FOR 0
1204 004024 001403          BEQ 10$          ;PASS
1205          004026 104000          ERROR                ;ERROR
          004030 004026          .-2                ;-----
1206          ;CCR DATA TEST
1207          ;WROTE 0 INTO CCR BIT07; READ 1
1208
1209 004032 000000          0
1210 004034 000240          10$:            NOP                ;END OF TEST
1211
1212
1213 004036 005257 001464          T20:            INC $TESTN          ;UPDATE TEST ID
          .:*****
          .SBTTL
          .SBTTL TEST 20
          .SBTTL
          .:*****
          .SBTTL WRITE ONE INTO CCR BIT07 THEN READ CCR
          .SBTTL IF CCR BIT07 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
          .SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
          .:*****
1219 004042 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          004044 004054          40$              ;ERROR/LOOP ON TEST
          004046 004054          1$              ;TEST START LOCATION
          004050 000000          0                ;LOOP ON ERROR START LOCATION
          004052 004066          25$             ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          004054          40$:            ;LOOP ON ERROR END LOCATION
          1220 004054 052737 000200 177746          1$:            BIS #BIT07,CCR          ;WRITE 1
          1221 004062 013700 177746          MOV CCR,RO          ;SAVE CCR CONTENTS
1222          25$:            NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
          004066 000240          NOP                ;FOR LOOP ON ERROR
          004070 000240
1223 004072 012737 001015 177746          MOV #OFF,CCR          ;DISABLE CACHE
1224 004100 032700 000200          BIT #BIT07,RO          ;CHECK FOR 1
1225 004104 001003          BNE 10$          ;PASS

```

1226	004106	104000	ERROR	:ERROR
	004110	004106	.-2	:-----
1227				:CCR DATA TEST
1228				:WROTE 1 INTO CCR BIT07; READ 0
1229				
1230	004112	000000		
1231	004114	00C240	108: NOP	:END OF TEST
1232				
1233				

```

1235 004116 005237 001464      T21:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 21
.SBTTL
1236
1237
1238
1239
1240
.SBTTL          WRITE ZERO INTO CCR BIT08(FC) THEN READ CCR
.SBTTL IF CCR BIT08 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
.SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
:*****
1241 004122 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:*****
:ERROR/LOOP ON TEST
004124 004134          40$          ;TEST START LOCATION
004126 004134          1$          ;LOOP ON ERROR START LOCATION
004130 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004132 004142          25$         ;LOOP ON ERROR END LOCATION
004134          40$:
1242 004134 042737 000400 177746      1$:          BIC #BIT08,CCR          ;WRITE 0
1243
004142 000240          25$:         NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
004144 000240          NOP          ;FOR LOOP ON ERROR

1244 004146 032737 000400 177746      BIT #BIT08,CCR          ;CHECK FOR 0
1245 004154 001401          BEQ 10$         ;PASS
1246
1247
1248
1249 004156 000000          0          ;CCR DATA TEST
1250 004160 000240          10$:         NOP          ;WROTE 0 INTO CCR BIT08; READ 1
1251
1252
1253 004162 005237 001464      T22:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 22
.SBTTL
1254
1255
1256
1257
1258
.SBTTL          WRITE 1 INTO CCR BIT09(UCB) AND ASSURE IT READS 1.
.SBTTL IF CCR BIT09 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
.SBTTL OR CACHE REGISTER DATA PATH MAY BE AT FAULT
:*****
1259 004166 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:*****
:ERROR/LOOP ON TEST
004170 004200          40$          ;TEST START LOCATION
004172 004200          1$          ;LOOP ON ERROR START LOCATION
004174 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004176 004206          25$         ;LOOP ON ERROR END LOCATION
004200          40$:
1260 004200 052737 001000 177746      1$:          BIS #UCB,CCR          ;
1261
004206 000240          25$:         NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
004210 000240          NOP          ;FOR LOOP ON ERROR

1262 004212 032737 001000 177746      BIT #BIT09,CCR          ;CHECK FOR 1
1263 004220 001003          BNE 10$         ;PASS
1264
004222 104000          ERROR          ;ERROR

```

```

        004224 004222          .-2          ;-----
1265                                     ;CCR DATA TEST
1266                                     ;WROTE 1 INTO CCR BIT09; READ 0
1267
1268 004226 000000          0
1269 004230 000240          10$: NOP          ;END OF TEST
1270
1271 004232 005237 001464  T23:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 23
.SBTTL
:*****
1272 .SBTTL WRITE ZERO INTO CCR BIT10(WWPT) AND READ 0
1273 .SBTTL
1274
1275 :*****
1276 004236 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          004240 004250          40$          ;ERROR/LOOP ON TEST
          004242 004250          1$          ;TEST START LOCATION
          004244 000000          0          ;LOOP ON ERROR START LOCATION
          004246 004256          25$          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          004250          40$:          ;LOOP ON ERROR END LOCATION
1277 004250 042737 002000 177746  1$: BIC #BIT10,CCR          ;WRITE 0
1278          004256 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          004260 000240          NOP          ;FOR LOOP ON ERROR
1279 004262 032737 002000 177746  BIT #BIT10,CCR          ;CHECK FOR 0
1280 004270 001403          BEQ 10$          ;PASS
1281
          004272 104000          ERROR          ;ERROR
          004274 004272          .-2          ;-----
1282                                     ;CCR DATA TEST
1283                                     ;WROTE 0 INTO CCR BIT 10; READ 1
1284
1285 004276 000000          0
1286 004300 000240          10$: NOP          ;END OF TEST
1287
1288
1289 .SBTTL *
1290 .SBTTL
1291 .SBTTL *          CACHE CONTROL CONTROL REGISTER UNUSED BIT TEST(CCR)
1292 .SBTTL
1293 .SBTTL *
1294
1295
1296 004302 005237 001464  T24:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 24
.SBTTL
:*****
1297 .SBTTL WRITE INTO UNUSED CCR REGISTER BIT01 THEN READ CCR
1298 .SBTTL IF CCR BIT01 READ AS ONE THEN CACHE DATA PATH
1299
    
```

```

1300 .SBTTL
1301 :*****
1302 004306 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          40$              ;ERROR/LOOP ON TEST
          1$              ;TEST START LOCATION
          0                ;LOOP ON ERROR START LOCATION
          25$             ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          25$             ;LOOP ON ERROR END LOCATION
1303 004320 052737 000002 177746 40$:
1304 004326 000240          1$:      BIS #BIT01,CCR          ;WRITE 1 INTO UNUSED BIT
          004330 000240          25$:     NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
          004330 000240          NOP                ;FOR LOOP ON ERROR
1305 004332 032737 000002 177746          BIT #BIT01,CCR          ;CHECK THAT BIT READS 0
1306 004340 001403          BEQ 10$          ;PASS
1307 004342 104000          ERROR                ;ERROR
          004344 004342          .-2              ;-----
1308                                     ;CCR UNUSED BIT TEST
1309                                     ;READ 1 FROM UNUSED CCR BIT01
1310                                     ;SHOULD READ 0
1311
1312 004346 000000          0
1313 004350 000240          10$:     NOP                ;END OF TEST
1314
1315
1316
1317 004352 005237 001464          T25:      INC $TESTN          ;UPDATE TEST ID

```

```

:*****
.SBTTL
.SBTTL TEST 25
.SBTTL
:*****
.SBTTL WRITE ONE INTO UNUSED CCR BIT04 THEN READ CCR
.SBTTL IF CCR BIT04 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
.SBTTL
:*****
1324 004356 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          40$              ;ERROR/LOOP ON TEST
          1$              ;TEST START LOCATION
          0                ;LOOP ON ERROR START LOCATION
          25$             ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          25$             ;LOOP ON ERROR END LOCATION
1325 004370 052737 000020 177746 40$:
1326 004376 000240          1$:      BIS #BIT04,CCR          ;WRITE 1 INTO UNUSED BIT
          004400 000240          25$:     NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
          004400 000240          NOP                ;FOR LOOP ON ERROR
1327 004402 032737 000020 177746          BIT #BIT04,CCR          ;CHECK THAT BIT READS 0
1328 004410 001403          BEQ 10$          ;PASS
1329 004412 104000          ERROR                ;ERROR
          ;-----

```

```

1330 004414 004412          .-2          ;CCR UNUSED BIT TEST
1331                                     ;READ 1 FROM UNUSED CCR BIT04
1332                                     ;SHOULD READ 0
1333
1334 004416 000000          0
1335 004420 000240          10$: NOP          ;END OF TEST
1336
1337
1338 004422 005237 001464  T26:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 26
.SBTTL
:*****
1339 .SBTTL WRITE ONE INTO UNUSED CCR BIT05 THEN READ CCR
1340 .SBTTL IF CCR BIT05 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
1341 .SBTTL
1342
1343
1344 004426 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:*****
004430 004440          40$          ;ERROR/LOOP ON TEST
004432 004440          1$          ;TEST START LOCATION
004434 000000          0          ;LOOP ON ERROR START LOCATION
004436 004446          25$          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004440          40$:          ;LOOP ON ERROR END LOCATION
1345 004440 052737 000040 177746 1$: BIT #BIT05,CCR          ;WRITE 1 INTO UNUSED BIT
1346          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HER
          004446 000240          NOP          ;FOR LOOP ON ERROR
          004450 000240

1347 004452 032737 000040 177746 BIT #BIT05,CCR          ;CHECK THAT BIT READS 0
1348 004460 001403          BEQ 10$          ;PASS
1349
          004462 104000          ERROR          ;ERROR
          004464 004462          .-2          ;-----

1350                                     ;CCR UNUSED BIT TEST
1351                                     ;READ 1 FROM UNUSED CCR BIT05
1352                                     ;SHOULD READ 0
1353
1354 004466 000000          0
1355 004470 000240          10$: NOP          ;END OF TEST
1356
1357
1358 004472 005237 001464  T27:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 27
.SBTTL
:*****
1359 .SBTTL WRITE ONE INTO UNUSED CCR BIT08 THEN READ CCR
1360 .SBTTL IF CCR BIT08 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
1361 .SBTTL
1362
1363
1364 004476 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:*****
          ;ERROR/LOOP ON TEST

```

```

004500 004510 40$ :TEST START LOCATION
004502 004510 1$ :LOOP ON ERROR START LOCATION
004504 000000 0 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004506 004516 25$ :LOOP ON ERROR END LOCATION
1365 004510 052737 000400 177746 40$:
1366 004510 052737 000400 177746 1$: BIS #BIT08,CCR ;WRITE 1 INTO UNUSED BIT
004516 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
004520 000240 NOP ;FOR LOOP ON ERROR
1367 004522 032737 000400 177746 BIT #BIT08,CCR ;CHECK THAT BIT READS 0
1368 004530 001403 BEQ 10$ ;PASS
1369 004532 104000 ERROR ;ERROR
004534 004532 .-2 ;-----
1370 004534 004532 .-2 ;CCR UNUSED BIT TEST
1371 ;READ 1 FROM UNUSED CCR BIT08
1372 ;SHOULD READ 0
1373
1374 004536 000000 0
1375 004540 000240 10$: NOP ;END OF TEST
1376
1377
1378 004542 005237 001464 T30: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 30
.SBTTL
;*****
1379 .SBTTL WRITE ONE INTO UNUSED CCR BIT11 THEN READ CCR
1380 .SBTTL IF CCR BIT11 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
1381 .SBTTL
1382
1383 ;*****
1384 004546 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
004550 004560 40$ ;ERROR/LOOP ON TEST
004552 004560 1$ ;TEST START LOCATION
004554 000000 0 ;LOOP ON ERROR START LOCATION
004556 004566 25$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004560 004566 ;LOOP ON ERROR END LOCATION
1385 004560 052737 004000 177746 40$:
1386 004560 052737 004000 177746 1$: BIS #BIT11,CCR ;WRITE 1 INTO UNUSED BIT
004566 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
004570 000240 NOP ;FOR LOOP ON ERROR
1387 004572 032737 004000 177746 BIT #BIT11,CCR ;CHECK THAT BIT READS 0
1388 004600 001403 BEQ 10$ ;PASS
1389 004602 104000 ERROR ;ERROR
004604 004602 . 2 ;-----
1390 ;CCR UNUSED BIT TEST
1391 ;READ 1 FROM UNUSED CCR BIT11
1392 ;SHOULD READ 0
1393
1394 004606 000000 0

```


1395 004610 000240
1396 004612 005237 001464

T31: 10\$: NOP ;END OF TEST
INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 31
.SBTTL

1397
1398
1399
1400
1401

:*****
WRITE ONE INTO UNUSED CCR BIT14 THEN READ CCR
.SBTTL IF CCR BIT14 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
.SBTTL

1402 004616 000004

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

004620 004630
004622 004630
004624 000000
004626 004636
004630

1403 004630 052737 040000 177746

40\$:
1\$: BIS #BIT14,CCR ;WRITE 1 INTO UNUSED BIT

1404
004636 000240
004640 000240

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

1405 004642 032737 040000 177746

BIT #BIT14,CCR ;CHECK THAT BIT READS 0
BEQ 10\$;PASS

1406 004650 001403

1407 004652 104000

ERROR ;ERROR
;-----

004654 004652

.-2
;CCR UNUSED BIT TEST
;READ 1 FROM UNUSED CCR BIT14
;SHOULD READ 0

1408
1409
1410
1411

1412 004656 000000
1413 004660 000240

0
10\$: NOP ;END OF TEST

1414
1415

1416 004662 005237 001464

T32: INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 32
.SBTTL

1417
1418
1419
1420
1421

:*****
WRITE 1 INTO UNUSED CCR BIT15 THEN READ CCR
.SBTTL IF CCR BIT15 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
.SBTTL

1422 004666 000004

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

004670 004700
004672 004700
004674 000000
004676 004706
004700

1423 004700 052737 100000 177746

40\$:
1\$: BIS #BIT15,CCR ;WRITE 1 INTO UNUSED BIT

1424
004706 000240

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

```

004710 000240          NOP          ;FOR LOOP ON ERROR
1425 004712 032737 100000 177746    BIT #BIT15,CCR          ;CHECK THAT BIT READS 0
1426 004720 001403          BEQ 10$          ;PASS
1427          004722 104000          ERROR          ;ERROR
          004724 004722          .-2          ;-----
1428          ;CCR UNUSED BIT TEST
1429          ;READ 1 FROM UNUSED CCR BIT15
1430          ;SHOULD READ 0
1431
1432 004726 000000          0
1433 004730 000240          10$: NOP          ;END OF TEST
1434
1435
1436          .SBTTL *
1437          .SBTTL
1438          .SBTTL *          CACHE MEMORY ERROR REGISTER UNUSED BIT TEST (CME)
1439          .SBTTL
1440          .SBTTL *
1441
1442
1443 004732 005237 001464    T33:          INC $TESTN          ;UPDATE TEST ID
          ;*****
          .SBTTL
          .SBTTL TEST 33
          .SBTTL
          ;*****
          .SBTTL ATTEMPT WRITE 1 INTO ALL UNUSED BITS OF CME.
          .SBTTL ALL BITS SHOULD READ 0.
          .SBTTL
          ;*****
1444
1445
1446
1447
1448
1449
1450 004736 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          004740 004750          40$          ;TEST START LOCATION
          004742 004750          1$          ;LOOP ON ERROR START LOCATION
          004744 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          004746 004756          25$          ;LOOP ON ERROR END LOCATION
          004750          40$:
1451 004750 052737 000001 177744    1$: BIS #BIT00,CME          ;WRITE 1 INTO BIT00
1452          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          004756 000240          NOP          ;FOR LOOP ON ERROR
          004760 000240
1453 004762 032737 000001 177744    BIT #BIT00,CME          ;CHECK FOR 0
1454 004770 001403          BEQ 10$          ;PASS
1455          004772 104000          ERROR          ;ERROR
          004774 004772          .-2          ;-----
1456          ;CME UNUSED BIT TEST
1457          ;READ 1 FROM UNUSED CME BIT00
1458 004776 000000          0
1459 005000 000240          10$: NOP          ;END OF TEST
1460 005002 005237 001464    T34:          INC $TESTN          ;UPDATE TEST ID

```

.SBTTL
.SBTTL TEST 34
.SBTTL

```

1461 005006 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          005010 005020          40$          ;ERROR/LOOP ON TEST
          005012 005020          1$          ;TEST START LOCATION
          005014 000000          0          ;LOOP ON ERROR START LOCATION
          005016 005026          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          005020          40$:          ;LOOP ON ERROR END LOCAT'ON
1462 005020 052737 000002 177744 1$:      BIS #BIT01,CME      ;WRITE 1 INTO BIT01
1463          005026 000240          25$:      NOP          ;INSTRUC JN 'JMP 1$' PLACED HERE
          005030 000240          NOP          ;FOR LO. ON ERROR
1464 005032 032737 000002 177744          BIT #BIT01,CME      ;CHECK FOR 0
1465 005040 001403          BEQ 10$          ;PASS
1466          005042 104000          ERROR          ;ERROR
          005044 005042          .-2          ;-----
1467          005046 000000          0          ;CME UNUSED BIT TEST
1468          005050 000240          10$:      NOP          ;READ 1 FROM UNUSED CME BIT01
1469          005052 005237 001464 T35:      INC $TESTN      ;END OF TEST
          005052 005237 001464          ;UPDATE TEST ID

```

.SBTTL
.SBTTL TEST 35
.SBTTL

```

1472 005056 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          005060 005070          40$          ;ERROR/LOOP ON TEST
          005062 005070          1$          ;TEST START LOCATION
          005064 000000          0          ;LOOP ON ERROR START LOCATION
          005066 005076          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          005070          40$:          ;LOOP ON ERROR END LOCATION
1473 005070 052737 000004 177744 1$:      BIS #BIT02,CME      ;WRITE 1 INTO BIT02
1474          005076 000240          25$:      NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          005100 000240          NOP          ;FOR LOOP ON ERROR
1475 005102 032737 000004 177744          BIT #BIT02,CME      ;CHECK FOR 0
1476 005110 001403          BEQ 10$          ;PASS
1477          005112 104000          ERROR          ;ERROR
          005114 005112          .-2          ;-----
1478          005116 000000          0          ;CME UNUSED BIT TEST
1479          005120 000240          10$:      NOP          ;READ 1 FROM UNUSED CME BIT02
1480          005122 005237 001464 T36:      INC $TESTN      ;END OF TEST
          005122 005237 001464          ;UPDATE TEST ID

```

.SBTTL
.SBTTL TEST 36

```

      .SBTTL
1483 005126 000004          .          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      005130 005140          40$          ;ERROR/LOOP ON TEST
      005132 005140          1$          ;TEST START LOCATION
      005134 000000          0          ;LOOP ON ERROR START LOCATION
      005136 005146          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      005140          40$:          ;LOOP ON ERROR END LOCATION
1484 005140 052737 000010 177744      1$:          BIS #BIT03,CME          ;WRITE 1 INTO BIT03
1485          005146 000240          25$:          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
      005150 000240          NOP          ;FOR LOOP ON ERROR
1486 005152 032737 000010 177744      BIT #BIT03,CME          ;CHECK FOR 0
1487 005160 001403          BEQ 10$          ;PASS
1488          005162 104000          ERROR          ;ERROR
      005164 005162          .-2          ;-----
1489          005164 005162          .-2          ;CME UNUSED BIT TEST
1490          ;READ 1 FROM UNUSED CME BIT03
1491 005166 000000          0
1492 005170 000240          10$:          NOP          ;END OF TEST
1493 005172 005237 001464      T37:          INC $TESTN          ;UPDATE TEST ID

```

```

*****
.SBTTL
.SBTTL TEST 37
.SBTTL

```

```

1494 005176 000004          .          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      005200 005210          40$          ;ERROR/LOOP ON TEST
      005202 005210          1$          ;TEST START LOCATION
      005204 000000          0          ;LOOP ON ERROR START LOCATION
      005206 005216          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      005210          40$:          ;LOOP ON ERROR END LOCATION
1495 005210 052737 000020 177744      1$:          BIS #BIT04,CME          ;WRITE 1 INTO BIT04
1496          005216 000240          25$:          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
      005220 000240          NOP          ;FOR LOOP ON ERROR
1497 005222 032737 000020 177744      BIT #BIT04,CME          ;CHECK FOR 0
1498 005230 001403          BEQ 10$          ;PASS
1499          005232 104000          ERROR          ;ERROR
      005234 005232          .-2          ;-----
1500          005234 005232          .-2          ;CME UNUSED BIT TEST
1501          ;READ 1 FROM UNUSED CME BIT04
1502 005236 000000          0
1503 005240 000240          10$:          NOP          ;END OF TEST
1504 005242 005237 001464      T40:          INC $TESTN          ;UPDATE TEST ID

```

```

*****
.SBTTL
.SBTTL TEST 40
.SBTTL

```

```

1505 005246 000004          .          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      ;ERROR/LOOP ON TEST

```

```

005250 005260 40$ :TEST START LOCATION
005252 005260 1$ :LOOP ON ERROR START LOCATION
005254 000000 0 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
005256 005266 25$ :LOOP ON ERROR END LOCATION
1506 005260 052737 000400 177744 40$:
1507 005266 000240 1$: BIS #BIT08,CME :WRITE 1 INTO BIT08
005270 000240 25$: NOP :INSTRUCTION 'JMP 1$' PLACED HERE
005272 032737 000400 177744 :NOP :FOR LOOP ON ERROR
1508 005272 032737 000400 177744 BIT #BIT08,CME :CHECK FOR 0
1509 005300 001403 BEQ 10$ :PASS
1510 005302 104000 ERROR :ERROR
005304 005302 .-2 :-----
1511 005304 005302 .-2 :CME UNUSED BIT TEST
1512 :READ 1 FROM UNUSED CME BIT08
1513 005306 000000 0
1514 005310 000240 10$: NOP :END OF TEST
1515 005312 005237 001464 T41: INC $TESTN :UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 41
.SBTTL
1516 005316 000004 SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
005320 005330 40$ :ERROR/LOOP ON TEST
005322 005330 1$ :TEST START LOCATION
005324 000000 U :LOOP ON ERROR START LOCATION
005326 005336 25$ :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
005330 :LOOP ON ERROR END LOCATION
1517 005330 052737 001000 177744 40$:
1518 005336 000240 1$: BIS #BIT09,CME :WRITE 1 INTO BIT09
005340 000240 25$: NOP :INSTRUCTION 'JMP 1$' PLACED HERE
005342 032737 001000 177744 :NOP :FOR LOOP ON ERROR
1519 005342 032737 001000 177744 BIT #BIT09,CME :CHECK FOR 0
1520 005350 001403 BEQ 10$ :PASS
1521 005352 104000 ERROR :ERROR
005354 005352 .-2 :-----
1522 005354 005352 .-2 :CME UNUSED BIT TEST
1523 :READ 1 FROM UNUSED CME BIT09
1524 005356 000000 0
1525 005360 000240 10$: NOP :END OF TEST
1526 005362 005237 001464 T42: INC $TESTN :UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 42
.SBTTL
1527 005366 000004 SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
005370 005400 40$ :ERROR/LOOP ON TEST
005372 005400 1$ :TEST START LOCATION
005374 000000 0 :LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST

```

```

005376 005406           25$           ;LOOP ON ERROR END LOCATION
005400
1528 005400 052737 002000 177744   40$:   BIS #BIT10,CME           ;WRITE 1 INTO BIT10
1529           1$:
005406 000240           25$:   NOP                       ;INSTRUCTION 'JMP 1$' PLACED HERE
005410 000240           NOP                       ;FOR LOOP ON ERROR
1530 005412 032737 002000 177744   BIT #BIT10,CME           ;CHECK FOR 0
1531 005420 001403           BEQ 10$                   ;PASS
1532 005422 104000           ERROR                       ;ERROR
                                ;-----
005424 005422           .-2
                                ;CME UNUSED BIT TEST
                                ;READ 1 FROM UNUSED CME BIT10
1533 005426 000000           0
1534 005430 000240           10$:  NOP                       ;END OF TEST
1535 005432 000240           INC $TESTN                ;UPDATE TEST ID
1536 005432 005237 001464   T43:
:*****
.SBTTL
.SBTTL TEST 43
.SBTTL
1538 005436 000004           SCPCND                     ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
005440 005450           40$:
005442 005450           1$:   ;TEST START LOCATION
005444 000000           0                           ;LOOP ON ERROR START LOCATION
005446 005456           25$:   ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
005450           ;LOOP ON ERROR END LOCATION
1539 005450 052737 004000 177744   40$:   BIS #BIT11,CME           ;WRITE 1 INTO BIT11
1540           1$:
005456 000240           25$:  NOP                       ;INSTRUCTION 'JMP 1$' PLACED HERE
005460 000240           NOP                       ;FOR LOOP ON ERROR
1541 005462 032737 004000 177744   BIT #BIT11,CME           ;CHECK FOR 0
1542 005470 001403           BEQ 10$                   ;PASS
1543 005472 104000           ERROR                       ;ERROR
                                ;-----
005474 005472           .-2
                                ;CME UNUSED BIT TEST
                                ;READ 1 FROM UNUSED CME BIT11
1544 005476 000000           0
1545 005500 000240           10$:  NOP                       ;END OF TEST
1546 005502 000240           INC $TESTN                ;UPDATE TEST ID
1547 005502 005237 001464   T44:
:*****
.SBTTL
.SBTTL TEST 44
.SBTTL
1549 005506 000004           SCPCND                     ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
005510 005520           40$:
005512 005520           1$:   ;TEST START LOCATION
005514 000000           0                           ;LOOP ON ERROR START LOCATION
005516 005526           25$:   ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
005520           ;LOOP ON ERROR END LOCATION
1550 005520 052737 010000 177744   40$:   BIS #BIT12,CME           ;WRITE 1 INTO BIT12
1550           1$:

```

```

1551      005526 000240      25$:  NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
          005530 000240      NOP      ;FOR LOOP ON ERROR

1552 005532 032737 010000 177744      BIT #BIT12,CME      ;CHECK FOR 0
1553 005540 001403      BEQ 10$      ;PASS
1554      005542 104000      ERROR      ;ERROR
          005544 005542      .-2      ;-----

1555      005546 000000      0      ;CME UNUSED BIT TEST
1556      005550 000240      NOP      ;READ 1 FROM UNUSED CME BIT12
1557 005546 000000      0
1558 005550 000240      10$:  NOP      ;END OF TEST
1559 005552 005237 001464      T45:  INC $TESTN ;UPDATE TEST ID
          ;*****
          .SBTTL
          .SBTTL TEST 45
          .SBTTL

1560 005556 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          005560 005570      40$      ;ERROR/LOOP ON TEST
          005562 005570      1$      ;TEST START LOCATION
          005564 000000      0      ;LOOP ON ERROR START LOCATION
          005566 005576      25$      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          005570      ;LOOP ON ERROR END LOCATION

1561 005570 052737 020000 177744      40$:  BIS #BIT13,CME ;WRITE 1 INTO BIT13
1562      005576 000240      1$:  ;
          005600 000240      25$:  NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
          ;FOR LOOP ON ERROR

1563 005602 032737 020000 177744      BIT #BIT13,CME      ;CHECK FOR 0
1564 005610 001403      BEQ 10$      ;PASS
1565      005612 104000      ERROR      ;ERROR
          005614 005612      .-2      ;-----

1566      005616 000000      0      ;CME UNUSED BIT TEST
1567      005620 000240      NOP      ;READ 1 FROM UNUSED CME BIT13
1568 005616 000000      0
1569 005620 000240      10$:  NOP      ;END OF TEST
1570 005622 005237 001464      T46:  INC $TESTN ;UPDATE TEST ID
          ;*****
          .SBTTL
          .SBTTL TEST 46
          .SBTTL

1571 005626 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          005630 005640      40$      ;ERROR/LOOP ON TEST
          005632 005640      1$      ;TEST START LOCATION
          005634 000000      0      ;LOOP ON ERROR START LOCATION
          005636 005646      25$      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          005640      ;LOOP ON ERROR END LOCATION

1572 005640 052737 040000 177744      40$:  BIS #BIT14,CME ;WRITE 1 INTO BIT14
1573      005646 000240      1$:  ;
          005650 000240      25$:  NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
          ;FOR LOOP ON ERROR
    
```

```

1574 005652 032737 040000 177744      BIT #BIT14,CME      ;CHECK FOR 0
1575 005660 001403                      BEQ 10$             ;PASS
1576      005662 104000                      ERROR              ;ERROR
                               ;-----
      005664 005662                      .-2
1577                                         ;CME UNUSED BIT JEST
1578                                         ;READ 1 FROM UNUSED CME BIT14
1579 005666 000000
1580 005670 000240      10$:      0              ;END OF TEST
                               NOP
1581
1582
1583      .SBTTL *
1584      .SBTTL
1585      .SBTTL *
1586      .SBTTL
1587      .SBTTL *
1588
1589
1590 005672 005237 001464      T47:      INC $TESTN      ;UPDATE TEST ID
                               ;*****
                               .SBTTL
                               .SBTTL TEST 47
                               .SBTTL
1591      ;*****
1592      .SBTTL REGISTER BYTE SELECTION LOGIC TEST
1593      .SBTTL WRITE ONE INTO LOW BYTE WRITE ZERO INTO HIGH BYTE
1594      .SBTTL VERIFY THAT LOW BYTE DATA IS NOT EFFECTED BY WRITE TO HIGH BYTE
1595      .SBTTL
1596      ;*****
1597 005676 000004                      SCPCND             ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                               ;ERROR/LOOP ON TEST
      005700 005710                      40$              ;TEST START LOCATION
      005702 005710                      1$               ;LOOP ON ERROR START LOCATION
      005704 000000                      0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      005706 005724                      25$             ;LOOP ON ERROR END LOCATION
      005710
1598 005710 152737 000004 177746      40$:      BISB #BIT02,CCR      ;WRITE 1 INTO CONTROL REGISTER BIT02
1599 005716 142737 000004 177747      1$:      BICB #BIT02,CCR+1    ;WRITE 0 INTO CONTROL REGISTER BIT10
1600
      005724 000240      25$:      NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
      005726 000240                      NOP              ;FOR LOOP ON ERROR
1601 005730 032737 000004 177746      BIT #BIT02,CCR      ;CHECK FOR 1
1602 005736 001003                      BNE 10$           ;PASS
1603
      005740 104000                      ERROR              ;ERROR
                               ;-----
      005742 005740                      .-2
1604                                         ;CACHE CONTROL REGISTER BYTE TESTS
1605                                         ;WROTE ONE INTO LOW BYTE BIT02
1606                                         ;WROTE ZERO INTO HIGH BYTE BIT10
1607                                         ;READ ZERO FROM BIT02
1608 005744 000000
1609 005746 000240      10$:      C              ;END OF TEST
                               NOP
1610

```

...

1611
1612 005750 005237 001464

T50: INC \$TESTN ;UPDATE TEST ID

.SBTTL
.SBTTL TEST 50
.SBTTL

1613
1614
1615
1616
1617

.SBTTL WRITE ZERO INTO HIGH BYTE WRITE ONE INTO LOW BYTE
.SBTTL VERIFY HIGH BYTE NOT EFFECTED BY WRITE INTO LOW BYTE
.SBTTL

1618 005754 000004

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

005756 005766
005760 005766
005762 000000
005764 006002
005766
1619 005766 142737 000004 177747
1620 005774 152737 000004 177746
1621

40\$:
\$: BICB #BIT02,CCR+1 ;WRITE 0 INTO CONTROL REGISTER BIT10
BISB #BIT02,CCR ;WRITE 1 INTO CONTROL REGISTER BIT02

006002 000240
006004 000240

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

1622 006006 032737 002000 177746
1623 006014 001403
1624

BIT #BIT10,CCR ;CHECK FOR 0 BIT10
BEQ 10\$;PASS

006016 104000
006020 006016

ERROR ;ERROR

.-2

1625
1626
1627
1628
1629 006022 000000
1630 006024 000240

;CACHE CONTROL REGISTER BYTE TESTS
;WROTE ZERO INTO HIGH BYTE BIT10
;WROTE ONE INTO LOW BYTE BIT02
;READ ZERO FROM BIT02 OR READ ONE FROM BIT10

10\$: NOP ;END OF TEST

1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643

.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *
CACHE MAINTENANCE REGISTER DATA TEST ((MR))

1644 006026 005237 001464

T51: INC \$TESTN ;UPDATE TEST ID

.SBTTL
.SBTTL TEST 51
.SBTTL

1645
1646
1647

.SBTTL VERIFY (MR BIT00(TDAR)) CAN BE WRITTEN TO A 0
.SBTTL


```

1677                                     ;MAINTENANCE REGISTER DATA TEST
1678                                     ;WROTE 1 INTO CMR BIT00; READ 0
1679 006156 000000
1680 006160 000240          10$:      0      NOP          ;END OF TEST
1681
1682
1683
1684 006162 005237 001464  T53:      INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 53
.SBTTL
:*****
1685 .SBTTL VERIFY CMR BIT01(HODO) CAN BE WRITTEN AS A 0.
1686 .SBTTL
1687
1688
1689 006166 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
006170 006200          40$          ;TEST START LOCATION
006172 006200          1$          ;LOOP ON ERROR START LOCATION
006174 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
006176 006212          25$          ;LOOP ON ERROR END LOCATION
006200          40$:
1690 006200 042737 000002 177750  1$:      BIC #BIT01,CMR          ;WRITE 0 INTO CMR BIT01
1691 006206 013700 177750          MOV CMR,R0          ;SAVE CONTENTS OF CMR
1692
006212 000240          25$:      NOP          ;INSTRUCTION 'JMP 1$' PLACED HFRE
006214 000240          NOP          ;FOR LOOP ON ERROR
1693 006216 005037 177750          CLR CMR          ;CLR MAINT
1694 006222 032700 000002          BIT #BIT01,R0          ;CHECK FOR 0 IN BIT01
1695 006226 001403          BEQ 10$          ;PASS
1696
006230 104000          ERROR          ;ERROR
006232 006230          .-2          ;-----
1697                                     ;MAINTENANCE REGISTER DATA TEST
1698                                     ;WROTE 0 INTO CMR BIT01; READ 1
1699 006234 000000
1700 006236 000240          10$:      0      NOP          ;END OF TEST
1701
1702
1703
1704 006240 005237 001464  T54:      INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 54
.SBTTL
:*****
1705 .SBTTL VERIFY CMR BIT01(HODO) CAN BE WRITTEN AS A 1.
1706 .SBTTL
1707
1708
1709 006244 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
006246 006256          40$          ;TEST START LOCATION
006250 006256          1$          ;LOOP ON ERROR START LOCATION
006252 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST

```

L⁴

```

006254 006270          25$:          ;LOOP ON ERROR END LOCATION
006256          40$:          ;WRITE 1 INTO CMR BIT01
1710 006256 052737 000002 177750 1$:  BIS #BIT01,CMR ;SAVE CONTENTS OF CMR
1711 006264 013700 177750          MOV CMR,R0
1712          25$:          ;INSTRUCTION 'JMP 1$' PLACED HERE
006270 000240          NOP          ;FOR LOOP ON ERROR
006272 000240          NOP

1713 006274 005037 177750          CLR CMR          ;CLR MAINT
1714 006300 032700 000002          BIT #BIT01,R0    ;CHECK FOR 1 IN BIT01
1715 006304 001003          BNE 10$        ;PASS
1716          ERROR          ;ERROR
                                ;-----

006310 006306          .-2          ;MAINTENANCE REGISTER DATA TEST
1717          0          ;WROTE 1 INTO CMR BIT01; READ 0
1718          0
1719 006312 000000          10$:  NOP          ;END OF TEST
1720 006314 000240
1721
1722
1723
1724 006316 005237 001464 T55:          INC $TESTN          ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 55
                                .SBTTL
1725          ;*****
1726          .SBTTL VERIFY CMR BIT03(AM) CAN BE WRITTEN AS A 0.
1727          .SBTTL
1728          ;*****
1729 006322 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
006324 006334          40$:          ;TEST START LOCATION
006326 006334          1$:          ;LOOP ON ERROR START LOCATION
006330 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
006332 006360          25$:          ;LOOP ON ERROR END LOCATION
006334          40$:          ;ALL 1'S TO AMR
1730 006334 012737 177777 177752 1$:  MOV #-1,CHR
1731 006342 112737 000374 177751      MOVB #374,CMR+1
1732 006350 105037 177750          CLRB CMR          ;WRITE 0 INTO CMR BIT03
1733 006354 013700 177750          MOV CMR,R0      ;SAVE CONTENTS OF CMR
1734          25$:          ;INSTRUCTION 'JMP 1$' PLACED HERE
006360 000240          NOP          ;FOR LOOP ON ERROR
006362 000240          NOP

1735 006364 005037 177750          CLR CMR          ;CLR MAINT
1736 006370 032700 000010          BIT #BIT03,R0    ;CHECK FOR 0 IN BIT03
1737 006374 001403          BEQ 10$        ;PASS
1738          ERROR          ;ERROR
                                ;-----

006400 006376          .-2          ;MAINTENANCE REGISTER DATA TEST
1739          0          ;WROTE 0 INTO CMR BIT03; READ 1
1740          0
1741 006402 000000          10$:  NOP          ;END OF TEST
1742 006404 000240

```

1743
1744
1745
1746 006406 005237 001464

T56: INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 56
.SBTTL

1747
1748
1749
1750

:*****
.SBTTL VERIFY BIT03(AM) CAN BE WRITTEN AS A 1.
.SBTTL
:*****

1751 006412 000004

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

006414 006424
006416 006424
006420 000000
006422 006456
006424
1752 006424 012737 177777 177750
1753 006432 112737 000374 177751
1754 006440 105037 177750
1755 006444 112737 000010 177750
1756 006452 013700 177750
1757

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

;PRECONDITION AM BIT TO 0
;WRITE 1 INTO AM BIT
;SAVE CONTENTS OF CMR

006456 000240
006460 000240

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

1758 006462 005037 177750
1759 006466 032700 000010
1760 006472 001003
1761

CLR CMR ;CLR MAINT
BIT #BIT03,RO ;CHECK FOR 1 IN BIT03
BNE 10\$;PASS

006474 104000
006476 006474

ERROR ;ERROR
.-2 ;-----

1762
1763
1764 006500 000000
1765 006502 000240
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777

;MAINTENANCE REGISTER DATA TEST
;WROTE 1 INTO CMR BIT03; READ 0

10\$: 0
NOP ;END OF TEST

.SBTTL CMR UNUSED BIT TESTS

1778
1779 006504 005237 001464

T57: INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 57
.SBTTL

```

1780
1781
1782
1783
1784
1785 006510 000004
      006512 006522
      006514 006522
      006516 000000
      006520 006530
      006522
1786 006522 052737 000040 177750
1787
      006530 000240
      006532 000240
1788 006534 032737 000040 177750
1789 006542 001403
1790
      006544 104000
      006546 006544
1791
1792
1793 006550 000000
1794 006552 000240
1795 006554 005237 001464
      T60:
      10$:
      40$:
      1$:
      25$:
      40$:
      1$:
      25$:
      10$:
      T61:
      10$:
      40$:
      1$:
      25$:
      10$:
      T61:
      10$:
      40$:
      1$:
      25$:
      10$:

```

```

:*****
.SBTTL ATTEMPT WRITE 1 INTO ALL UNUSED BITS OF CMR. ALL
.SBTTL BITS SHOULD READ 0.
.SBTTL
:*****

```

```

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40$ ;ERROR/LOOP ON TEST
1$ ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40$: ;LOOP ON ERROR END LOCATION
1$: BIS #BIT05,CMR ;WRITE 1 INTO BIT05
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIT #BIT05,CMR ;CHECK FOR 0
BEQ 10$ ;PASS
ERROR ;ERROR
.-2 ;-----
;CMR UNUSED BIT TEST
;READ 1 FROM UNUSED CMR BIT05
0
NOP ;END OF TEST
INC $TESTN ;UPDATE TEST ID

```

```

:*****
.SBTTL
.SBTTL TEST 60
.SBTTL

```

```

SCPCND ;SCOPE CONDITIONS:GO SET JP FOR LOOP ON
40$ ;ERROR/LOOP ON TEST
1$ ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40$: ;LOOP ON ERROR END LOCATION
1$: BIS #BIT06,CMR ;WRITE 1 INTO BIT06
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIS #BIT06,CMR ;CHECK FOR 0
BEQ 10$ ;PASS
ERROR ;ERROR
.-2 ;-----
;CMR UNUSED BIT TEST
;READ 1 FROM UNUSED CMR BIT02
0
NOP ;END OF TEST
INC $TESTN ;UPDATE TEST ID

```

```

:*****
.SBTTL

```

.SBTTL TEST 61
.SBTTL

```
1807 006630 000004          SPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          006632 006642          40$          ;ERROR/LOOP ON TEST
          006634 006642          1$          ;TEST START LOCATION
          006636 000000          0          ;LOOP ON ERROR START LOCATION
          006640 006650          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          006642          40$:         ;LOOP ON ERROR END LOCATION
1808 006642 052737 000200 177750 1$:      BIS #BIT07,CMR    ;WRITE 1 INTO BIT07
1809          006650 000240          25$:      NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          006652 000240          NOP          ;FOR LOOP ON ERROR
1810 006654 032737 000200 177750          BIT #BIT07,CMR    ;CHECK FOR 0
1811 006662 001403          BEQ 10$         ;PASS
1812          006664 104000          ERROR          ;ERROR
          006666 006664          .-2          ;-----
1813          ;CMR UNUSED BIT TEST
1814          ;READ 1 FROM UNUSED CMR BIT03
1815 006670 000000          0
1816 006672 000240          10$:      NOP          ;END OF TEST
```

1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831

.SBTTL *
.SBTTL
.SBTTL *
.SBTTL
.SBTTL *

ADDRESS MATCH REGISTER TESTS (AMR TESTS)

1832 006674 005237 001464

```
T62:          INC $TESTN          ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 62
.SBTTL
.SBTTL
.SBTTL MA<21:0> ADDRESS LINES ALL 1'S
.SBTTL CA<21:0> ADDRESS LINES ALL 0'S
.SBTTL AMR<21:0> DATA LINES ALL 0'S
:::          AM BIT SHOULD READ 1.
:::          -----
:*****
```

1833
1834
1835
1836
1837
1838
1839

```
1840 006700 000004          SPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          006702 006712          40$          ;ERROR/LOOP ON TEST
          006704 006712          1$          ;TEST START LOCATION
          006706 000000          0          ;LOOP ON ERROR START LOCATION
          006710 006752          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          ;LOOP ON ERROR END LOCATION
```

```
1841 006712 012737 177777 177752 40$: MOV #-1,CHR ;ALL 1'S TO AMR
1842 006720 112737 000374 177751 1$: MOVB #374,CMR+1 ;PRECONDITION AM BIT TO 0
1843 006726 105037 177750 CLR B CMR ;ALL 0'S TO AMR<21:0>
1844 006732 005037 177752 CLR CHR
1845 006736 105037 177751 CLR B CMR+1
1846 006742 005737 000000 TST 0 ;PLACE ALL 0'S ON CA<21:0>.HOWEVER,
1847 ;THIS IS NOT WHEN THE AM BIT IS SET:
1848 ;WHEN PAX ADDRESS LINES ARE NOT BEING
1849 ;ACCESSED BY THE CPU, THE CACHE DEFAULTS
1850 ;TO SELECTING MA<21:0> ADDRESS LINES.
1851 ;IN THIS SITUATION, MA<21:0> DEFAULTS
1852 ;TO ALL 1'S THEREBY PLACING ALL 0'S
1853 ;ON CA<21:0>. THEREFORE, FOLLOWING THE LOADING
1854 ;OF ALL 0'S INTO AMR<21:0>,AND BEFORE THE
1855 ;'TST 0' INSTRUCTION, THE AM BIT SHOULD
1856 ;BE SET DUE TO MATCH BETWEEN AMR<21:0> AND CA<21:0>
1857 ; ADDRESS LINES.
1858 006746 013703 177750 MOV CMR,R3 ;SAVE AM BIT RESULT IN CMR
1859 ;INSTRUCTION 'JMP 1$' PLACED HERE
          006752 000240 25$: NOP ;FOR LOOP ON ERROR
          006754 000240 NOP
1860 006756 032703 000010 BIT #AM,R3 ;AM BIT SHOULD READ 1 INDICATING MATCH
1861 006762 001003 BNE 10$ ;PASS
1862 006764 104000 ERROR ;ERROR
          006766 006764 .-2 ;-----
1863 ;AMR TESTS
1864 ;AMR BIT DID NOT READ 1 INDICATING
1865 ;A MATCH OF ALL 0'S BETWEEN MA TO CA<21:0>
1866 ;ADDRESS LINES AND AMR<21:0> DATA
1867 006770 000000 0 ;ERROR PRINT TERMIN.
1868 006772 000240 10$: NOP ;END OF TEST
1869
1870
1871 ;::: MEMORY MANAGEMENT
1872
1873 006774 012737 077406 172300 MAGPRE: MOV #77406,KPDR0 ;ALLOW ALL ACCESS TO KERNEL PAGE 0
1874 007002 012737 077406 172302 MOV #77406,KPDR1 ;ALLOW ALL ACCESS TO KERNEL PAGE 1
1875 007010 012737 077406 172304 MOV #77406,KPDR2 ;ALLOW ALL ACCESS TO KERNEL PAGE 2
1876 007016 012737 077406 172306 MOV #77406,KPDR3 ;ALLOW ALL ACCESS TO KERNEL PAGE 3
1877 007024 012737 077406 172310 MOV #77406,KPDR4 ;ALLOW ALL ACCESS TO KERNEL PAGE 4
1878 007032 012737 077406 172312 MOV #77406,KPDR5 ;ALLOW ALL ACCESS TO KERNEL PAGE 5
1879 007040 012737 077406 172314 MOV #77406,KPDR6 ;ALLOW ALL ACCESS TO KERNEL PAGE 6
1880 007046 012737 077406 172316 MOV #77406,KPDR7 ;ALLOW ALL ACCESS TO KERNEL PAGE 7
1881 007054 012737 000000 172340 MOV #0,KPAR0 ;MAP PAGE 0 FOR 0-4K
1882 007062 012737 000200 172342 MOV #200,KPAR1 ;MAP PAGE 1 FOR 4-8K
1883 007070 012737 000400 172344 MOV #400,KPAR2 ;MAP PAGE 2 FOR 8-12K
1884 007076 012737 000600 172346 MOV #600,KPAR3 ;MAP PAGE 3 FOR 12-16K
1885 007104 012737 177600 172356 MOV #177600,KPAR7 ;MAP PAGE 7 FOR 124-128K
1886
1887
1888 ; UNIBUS MAP
1889
```


1891 007112 012737 000000 170200
1892 007120 012737 000000 170202
1893 007126 012737 020000 170204
1894 007134 012737 000000 170206
1895 007142 012737 040000 170210
1896 007150 012737 000000 170212
1897 007156 012737 060000 170214
1898 007164 012737 000000 170216
1899
1900
1901
1902
1903

MOV #0,UMPRO0 ;MAP REGISTER SET 0 FOR 0-4K
MOV #0,UMPRO1 ;MAP REGISTER SET 1 FOR 4K-8K
MOV #20000,UMPRO2 ;MAP REGISTER SET 2 FOR 8K-12K
MOV #0,UMPRO3 ;MAP REGISTER SET 3 FOR 12K-16K
MOV #40000,UMPRO4
MOV #0,UMPRO5
MOV #60000,UMPRO6
MOV #0,UMPRO7

1904 007172 005237 001464

T63: INC \$TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 63
.SBTTL
.SBTTL
.SBTTL MA<21:0> ADDRESS LINES ALL 0'S
.SBTTL CA<21:0> ADDRESS LINES ALL 1'S
.SBTTL AMR<21:0> DATA LINES ALL 1'S
.SBTTL AM BIT SHOULD READ 1.
:*****

1905
1906
1907
1908
1909
1910

```
1912 007176 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$
                                0
                                25$
007200 007210          40$
007202 007210          1$
007204 000000          0
007206 007234          25$
007210
1913 007210 012737 177777 177752      40$:      MOV #-1,CHR          ;LOAD AMR<15:0> WITH 1'S FROM CHR<15:0>
1914 007216 112737 000374 177751      1$:      MOVB #374,CMR+1      ;LOAD AMR<21:16> ALL 1'S
1915 007224 105037 177750          CLR B CMR          ;PRECONDITION AM BIT TO 0
1916 007230 105737 177777          TST B 177777      ;PUT ALL 1'S ON MA AND PA ADDRESS LINES
1917
1918
                                25$:      NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP          ;FOR LOOP ON ERROR
1919 007240 032737 000010 177750      BIT #AM,CMR      ;AM BIT SHOULD READ 1 INDICATING MATCH
1920 007246 001003          BNE 10$          ;PASS
1921
                                ERROR          ;ERROR
                                .-----
007250 104000
007252 007250          .-2
1922
1923
1924
1925
1926 007254 000000          0          ;AMR TESTS
1927 007256 000240          10$:      NOP          ;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.
                                ;END OF TEST
1928
1929
1930 007260 005237 001464      T64:      INC $TESTN          ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 64
                                .SBTTL
                                ;*****
                                ;::: VERIFY NO AMR LINES SHORTED TO EACH OTHER; OR NO AMR LINES
                                ;::: SHORTED TO CA LINES
                                ;:::
                                .SBTTL
                                .SBTTL MA<21:0> ADDRESS LINES ALL 1'S
                                .SBTTL CA<21:0> ADDRESS LINES ALL 0'S
                                .SBTTL AMR<15:0> FLOATING 1 PATTERN
                                .SBTTL FOR EACH FLOATING 1 PATTERN AM BIT SHOULD READ 0.
                                .SBTTL
                                ;*****
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941 007264 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
                                ;LOOP ON ERROR START LOCATION
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                ;LOOP ON ERROR END LOCATION
                                40$
                                1$-40$+60000-14
                                0
                                25$-40$+60000-14
007266 007276          40$:      MOV #OFF,CCR          ;DISABLE CACHE
007270 060006          JSR R4,RELCTL      ;LOCATE TEST CODE TO LOW CACHE SPACE
007272 000000          10$:+2          ;ADDRESS OF START OF NEXT TEST
007274 060034
007276 012737 001015 177746
007304 004437 002324
007310 007456
```

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

```

1942 007312 012737 000001 002056          MOV #1,CHRPAT          ;SETUP 1ST PATTERN FOR CHR<15:0>
1943 007320                                     1$:
1944 007320 013737 002056 177752          MOV CHRPAT,CHR        ;LOAD AMR<15:0> FROM CHR<15:0>
1945 007326 105037 177751                  CLRB CMR+1            ;LOAD AMR<21:16> FROM CMR<15:10>
1946 007332 105037 177750                  CLRB CMR              ;PRECONDITION AM BIT TO 0
1947 007336 005737 000000                  TST 0                 ;PLACE ALL 0'S ON CA<21:0>.
1948                                         ;WHEN PAX ADDRESS LINES ARE NOT BEING
1949                                         ;ACCESSED BY THE CPU, THE CACHE DEFAULTS
1950                                         ;TO SELECTING MA<21:0> ADDRESS LINES.
1951                                         ;IN THIS SITUATION, MA<21:0> DEFAULTS
1952                                         ;TO ALL 1'S THEREBY PLACING ALL 0'S
1953                                         ;ON CA<21:0>. THEREFORE, FOLLOWING THE LOADING
1954                                         ;OF ALL 0'S INTO AMR<21:0>, AND BEFORE THE
1955                                         ;'TST 0' INSTRUCTION, ALL 0'S ARE PLACED
1956                                         ; ON CA<21:0> ADDRESS LINES.
1957 007342 013703 177750                  MOV CMR,R3           ;SAVE AM BIT RESULT IN CMR
1958 007346 000240                                     25$: NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
      007350 000240                                     NOP                  ;FOR LOOP ON ERROR

1959 007352 032703 000010                  BIT #AM,R3           ;CHECK FOR 0.
1960 007356 001432                          BEQ 9$                ;PASS
1961 007360 005037 047302                  CLR CA210+2         ;PREPARE CA210 FOR TYPEOUT
1962 007364 005037 047300                  CLR CA210
1963 007370 013737 002056 047306          MOV CHRPAT,AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>
1964                                         ;FOR ERROR TYPEOUT
1965 007376 013737 002056 047304          MOV CHRPAT,AMR210
1966 007404 012737 000017 002052          MOV #15,LOOP
1967 007412 006237 047304                                     3$: ASR AMR210
1968 007416 042737 100000 047304          BIC #100000,AMR210
1969 007424 005337 002052                  DEC LOOP
1970 007430 001370                          BNE 3$
1971 007432 104000                          ERROR                 ;ERROR
      007434 007432                          .-2                   ;-----

1972                                         ;AMR TESTS
1973                                         ;AM BIT SHOULD HAVE READ 0 INDICATING A
1974                                         ;NO-MATCH CONDITION.
1975 007436 047300                          CA210                 ;PRINT PATTERN USED FOR CACHE ADDRESS LINES CA<21:0>
1976 007440 047304                          AMR210                ;PRINT FLOATING 1 PATTERN USED FOR AMR<21:0> DATA
1977 007442 000000                          0
1978                                         ;:
1979 007444 006337 002056                                     9$: ASL CHRPAT          ;NEXT FLOATING 1 PATTERN
1980 007450 103401                          BCS 10$               ;IF PHYSICAL ADDRESS 100000 DONE; THEN FINISHED
1981 007452 000722                          BR 1$                 ;IF NOT CONTINUE WITH NXT PATTERN
1982 007454 000240                                     10$: NOP              ;END OF TEST
1983
1984
1985 007456 005237 001464          T65: INC $TESTN      ;UPDATE TEST ID
      .....
      .SBTTL
      .SBTTL TEST 65
      .SBTTL
      .SBTTL
1986
1987      .SBTTL MA<21:0> ADDRESS LINES ALL 1'S
    
```

```
1988 .SBTTL CA<21:0> ADDRESS LINES ALL 0'S
1989 .SBTTL AMR<21:16> FLOATING 1 PATTERN
1990 .SBTTL FOR EACH FLOATING 1 PATTERN AM BIT SHOULD READ 0.
1991 ;*****
1992 007462 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
007464 007474 40$ ;TEST START LOCATION
007466 060006 1$-40$+60000-14 ;LOOP ON ERROR START LOCATION
007470 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
007472 060034 25$-40$+60000-14 ;LOOP ON ERROR END LOCATION
007474 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
007502 004437 002324 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
007506 007636 10$+2 ;ADDRESS OF START OF NEXT TEST

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

1993 007510 012737 000004 002054 1$: MOV #4,CMRPAT ;SETUP 1ST PATTERN FOR CMR<21:16>
1994 007516 MOV#B CMRPAT,CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>
1995 007516 113737 002054 177751 CLR CHR ;LOAD ALL 0'S TO AMR<15:0> FROM CHR<15:0>
1996 007524 005037 177752 CLR#B CMR ;PRECONDITION AM BIT TO 0
1997 007530 105037 177750 TST 0 ;SAVE CMR CONTENTS. BEFORE THE FETCH
1998 007534 005737 000000 ;OF THIS INSTRUCTION, ALL 0'S WILL
1999 ;BE PLACED ON CA<21:0> LINES.
2000 MOV CMR,R3 ;SAVE CMR CONTENTS
2001 007540 013703 177750
2002 007544 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
007546 000240 NOP ;FOR LOOP ON ERROR

2003 007550 032703 000010 BIT #AM,R3 ;CHECK FOR 0.
2004 007554 001420 BEQ 9$ ;PASS
2005 007556 005037 047302 CLR CA210+2 ;PREPARE CA210 FOR PRINTOUT
2006 007562 005037 047300 CLR CA210
2007 007566 005037 047306 CLR AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>
2008 007572 013737 002054 047304 MOV CMRPAT,AMR210
2009 007600 006237 047304 ASR AMR210
2010 007604 104000 ERROR ;ERROR
;-2
2011 007606 007604 .-2 ;AMR TESTS
2012 ;AM BIT SHOULD HAVE READ 0 INDICATING A
2013 ;NO-MATCH CONDITION.
2014 007610 047300 CA<10 ;PRINT PATTERN USED FOR CACHE ADDRESS LINES CA<21:0>
2015 007612 047304 AMR210 ;PRINT FLOATING 1 PATTERN USED FOR AMR<21:0> DATA
2016 007614 000000 0
2017 007616 006337 002054 9$: ASL CMRPAT ;NEXT FLOATING 1 PATTERN
2018 007622 032737 000400 002054 BIT #400,CMRPAT ;IF PHYSICAL ADDRESS 10000000 DONE;FINISHED
2019 007630 001001 BNE 10$ ;IF NOT CONTINUE WITH NXT PATTERN
2020 007632 000731 BR 1$
2021 007634 000240 10$: NOP ;END OF TEST
2022
2023
2024 007636 005237 001464 T66: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
```

```
.SBTTL TEST 66
.SBTTL
.SBTTL
2025
2026
2027
2028
2029
2030
2031
2032 007642 000004
      007644 007654
      007646 060020
      007650 000000
      007652 060114
      007654 012737 001015 177746
      007662 004437 002324
      .007666 010102
      SCPCND
      40$
      1$-40$+60000-14
      0
      25$-40$+60000-14
      40$: MOV #OFF,CCR
      JSR R4,RELCTL
      10$+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

2033 007670 012737 000001 047332
2034 007676 012701 100001
2035
2036 007702 012737 170000 172350
2037
2038
2039
2040
2041
2042 007710 012737 000001 177572
2043 007716 012737 000020 172516
2044 007724 013737 047332 177752
2045 007732 105037 177751
2046 007736 105037 177750
2047 007742 023727 047332 000001
2048 007750 001004
2049 007752 105711
2050 007754 000240
2051 007756 000240
2052 007760 000403
2053 007762 005711
2054 007764 000240
2055 007766 000240
2056 007770 013703 177750
2057 007774 005037 177572
2058 010000 005037 172516
2059
      010004 000240
      010006 000240
      1$: MOV #1,SRO
      MOV #20,SR3
      MOV FLTPAT,CHR
      CLRB CMR+1
      CLRB CMR
      CMP FLTPAT,#1
      BNE 2$
      TSTB (R1)
      :
      NOP
      NOP
      BR 4$
      2$: TST (R1)
      :
      NOP
      NOP
      4$: MOV CMR,R3
      CLR SRO
      CLR SR3
      :DISABLE MEM. MNGMNT.
      25$: NOP
      NOP
      :INSTRUCTION 'JMP 1$' PLACED HERE
      :FOR LOOP ON ERROR

2060 010010 032703 000010
2061 010014 001017
2062 010016 013737 047332 047306
2063
2064 010024 005037 047304
      BIT #AM,R3
      BNE 9$
      MOV FLTPAT,AMR210+2
      CLR AMR210
      :CHECK FOR 1
      :PASS
      :PREPARE PATTERN USED FOR AMR<21:0>
      :FOR ERROR TYPEOUT
```

```
2065 010030 013737 047332 047302      MOV FLTPAT,CA210+2      ;PREPARE PATT. USED FOR CA<21:0 > FOR ERROR TYPE
2066 010036 005037 047300      CLR CA210
- 2067      010042 104000      ERROR                  ;ERROR
      010044 010042      .-2                    ;-----
2068      .AMR TESTS
2069      .AM BIT DID NOT READ 1
2070 010046 047300      CA210                  ;PRINT FLOATING 1 PATTERN USED FOR CA<21:0>
2071 010050 047304      AMR210                 ;PRINT FLOAT 1 PATRN. USED FOR AMR<21:0>
2072 010052 000000      0
2073 010054 006337 047332      9$: ASL FLTPAT           ;NXT PATTERN
2074 010060 032737 020000 047332  BIT #20000,FLTPAT     ;IF PATTERN 10000 DONE; FINISHED
2075 010066 001004      BNE 10$
2076 010070 006301      ASL R1                 ;IF NOT NEXT PASS
2077 010072 052701 100000      BIS #100000,R1
2078 010076 000704      BR 1$
2079 010100 000240      10$: NOP              ;END OF TEST
2080
2081
2082 010102 005237 001464      T67: INC $TESTN        ;UPDATE TEST ID
      .:*****
      .SBTTL
      .SBTTL TEST 67
      .SBTTL
2083      .SBTTL
2084      .SBTTL MA<21:13> ADDRESS LINES FLOATING 0
2085      .SBTTL CA<21:13> ADDRESS LINES FLOATING 1
2086      .SBTTL AMR<21:13> FLOAT. 1 PATRN.
2087      .SBTTL AM BIT READS 1
2088      .:*****
2089 010106 000004      SCPCND                 ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      010110 010120      40$                   ;ERROR/LOOP ON TEST
      010112 060036      1$-40$+60000-14     ;TEST START LOCATION
      010114 000000      0                          ;LOOP ON ERROR START LOCATION
      010116 060144      25$-40$+60000-14     ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      010120 012737 001015 177746 40$: MOV #OFF,CCR        ;LOOP ON ERROR END LOCATION
      010126 004437 002324      JSR R4,RELCTL          ;DISABLE CACHE
      010132 010460      10$+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
2090 010134 012737 171000 172350      2$: MOV #171000,KPAR4   ;MAP PAGE 4 FOR TOP 124K ADDRESSING
2091      ;WILL ALSO SELECT UNIBUS MAP REGISTER
2092      ;SET #4
2093
2094 010142 012737 000040 047332      MOV #40,FLTPAT         ;SETUP 1ST PATTERN FOR UMPRO9
2095 010150 012737 020000 170220      MOV #20000,UMPRO8     ;SETUP 1ST PATTERN FOR UMPRO8
2096      ;ACCESSING TOP 124K,AND ENABLING
2097      ;UNIBUS MAP, THE 1ST ADDRESS WILL BE
2098      ;CONSTRUCTED THRU THE PA<21:0>
2099      ;LINES AS 17020000 AND AS 00020000
2100      ;THRU THE MA<21:0> LINES. DUE TO TOP 124K
2101      ;ADDRESSING CA<21:0> WILL SELECT THE MA LINES.
2102 010156 012737 000200 002054      MOV #200,CMRPAT       ;SETUP 1ST PATTERN FOR CMR<15:0>
```

```

2103 010164 012737 020000 002056
2104 010172 012737 060126 000004
2105 010200 012737 000340 000006
2106 010206 113737 047333 170222
2107 010214 113737 002055 177751
2108 010222 013737 002056 177752
2109 010230 012737 000001 177572
2110 010236 012737 000060 172516
2111 010244 105037 177750
2112 010250 005737 100000
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125 010254 000240
2126 010256 000240
2127 010260 000401
2128 010262 022626
2129 010264 013703 177750
2130 010270 005037 177572
2131 010274 005037 172516
2132
    010300 000240
    010302 000240
2133 010304 032703 000010
2134 010310 001040
2135 010312 012737 000006 000004
2136 010320 005037 000006
2137
2138 010324 013737 002056 047306
2139 010332 013737 002056 047302
2140 010340 013737 047332 047304
2141 010346 013737 047332 047300
2142 010354 012737 000007 002052
2143 010362 006237 047300
2144 010366 006237 047304
2145 010372 005337 002052
2146 010376 001371
2147
    010400 104000
    010402 010400
2148
2149
2150 010404 047300
2151 010406 047304
2152 010410 000000
2153

```

```

1$: MOV #20000,CHRPAT
    MOV #3$-2$+60000,4
    MOV #340,6
    MOV FLTPAT+1,UMPRO9
    MOV CMRPAT+1,CMR+1
    MOV CHRPAT,CHR
    MOV #1,SRO
    MOV #60,SR3
    CLRB CMR
    TST 100000

3$: CMP (SP)+,(SP)+
4$: MOV CMR,R3
    CLR SRO
    CLR SR3

25$: NOP
     NOP

BIT #AM,R3
BNE 9$
MOV #6,4
CLR 6

6$: MOV CHRPAT,AMR210+2
    MOV CHRPAT,CA210+2
    MOV FLTPAT,AMR210
    MOV FLTPAT,CA210
    MOV #7,LOOP
    ASR CA210
    ASR AMR210
    DEC LOOP
    BNE 6$

ERROR
.-2

CA210
AMR210
0

```

```

;SETUP 1ST PATTERN FOR CHR<15:0>
;
;LOAD UPPER BITS OF UNIBUS MAP REGISTER
;LOAD AMR<21:16> FROM CMR<15:10>
;LOAD AMR<15:0> FROM CHR<15:0>
;ENABLE MEM MNGMENT
;ENABLE UNIBUS MAP AND 22-BIT MAPPING
;PRECONDITION AM BIT WITH 0
;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

;NO TRAP
;
;SAVE CMR CONTENTS

;DISABLE UNIBUS MAP

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

;CHECK FOR 1
;PASS

;PREPARE AMR210 AND CA210 FOR PRINTOUT

;ERROR
;-----

;AMR TESTS
;AM BIT DIT NOT READ 1
;PRINT CA<21:0> PATTERN USED
;PRINT AMR<21:0> PAT. USED

```

:::

```
2154 010412 006337 047332          9$: ASL FLTPAT          ;NEXT PATTERN FOR UMPRO1
2155 010416 032737 040000 047332    BIT #40000,FLTPAT      ;IF ADDRESS PATTERN 10000000 DON; FINISHED
2156 010424 001007                    BNE 8$
2157 010426 006337 170220          ASL UMPRO8          ;NEXT PATTERN FOR UMPRO8
2158 010432 006337 002054          ASL CMRPAT          ;NEXT PATTERN FOR CMR<15:10>
2159 010436 006337 002056          ASL CHRPAT          ;NEXT PATTERN FOR CHR<15:0>
2160 010442 000653                    BR 1$
2161 010444 012737 000006 000004    8$: MOV #6,4          ;RESTORE VECTORS
2162 010452 005037 000006          CLR 6
2163 010456 000240          10$: NOP              ;END OF TEST
2164
2165 010460 005237 001464          T70: INC $TESTN      ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 70
.SBTTL
.SBTTL
2166
2167
2168
2169
2170
2171
2172
2173
2174 010464 000004          ;: VERIFY NO STUCK 1 OR 0 PA LINES; OR NO PA LINES
;: STUCK TO EACH OTHER.
.SBTTL PA<14:0> ADDRESS LINES FLOATING 1
.SBTTL CA<14:0> ADDRESS LINES FLOATING 1
.SBTTL AMR<14:0> FLOATING 1 PATTERN
.SBTTL AM BIT READS 1
;*****
SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40$            ;TEST START LOCATION
010466 010476          1$-40$+60000-14    ;LOOP ON ERROR START LOCATION
010470 060004          0                  ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
010472 000000          25$-40$+60000-14  ;LOOP ON ERROR END LOCATION
010474 060034          40$: MOV #OFF,CCR   ;DISABLE CACHE
010476 012737 001015 177746    JSR R4,RELCTL      ;LOCATE TEST CODE TO LOW CACHE SPACE
010504 004437 002324          10$+2             ;ADDRESS OF START OF NEXT TEST
010510 010626

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

2175 010512 012701 000001          1$: MOV #1,R1        ;R1 CONTAINS 1ST FLOATING 1 PATTERN: 000001
2176 010516 010137 177752          MOV R1,CHR         ;LOAD AMR<15:0> FROM CHR<15:0>
2177 010522 105037 177751          CLRB CMR+1        ;LOAD AMR<21:16> FROM CMR<15:10>
2178 010526 105037 177750          CLRB CMR          ;PRECONDITION AM BIT TO 0
2179 010532 022701 000001          CMP #1,R1         ;IF PATTERN IS 000001 USE TSTB
2180 010536 001002                    BNE 2$
2181 010540 105711                    TSTB (R1)
2182 010542 000401                    BR 3$
2183 010544 005711          2$: TST (R1)       ;READ ADDRESS SPECIFIED IN R1
;WHICH WILL PLACE FLOATING 1 PATTERN ON ADDRESS LINE
;PA WILL BE SELECTED TO FEED CA LINES.
2184
2185
2186 010546          3$:
2187
010546 000240          25$: NOP           ;INSTRUCTION 'JMP 1$' PLACED HERE
010550 000240          NOP              ;FOR LOOP ON ERROR

2188 010552 032737 000010 177750    BIT #AM,CMR        ;CHECK FOR 1
2189 010560 001015                    BNE 9$            ;PASS
2190 010562 010137 047306          MOV R1,AMR210+2   ;PREPARE PATTERN USED FOR AMR<21:0>
2191
;FOR ERROR TYPE.
```



```

2192 010566 005037 047304      CLR AMR210
2193 010572 010137 047302      MOV R1,CA210+2      ;PREPARE PATTREN USED FOR CA<21:0>
2194 010576 005037 047300      CLR CA210
2195                               ERROR      ;ERROR
                               .-2      ;-----
2196                               ;AMR TESTS
2197                               ;AM BIT SHOULD HAVE READ 1 INDICATING A
2198                               ;MATCH CONDITION.
2199 010606 047300      CA210      ;PRINT FLOATING 1 PATTERN USED FOR
2200                               ;CACHE ADDRESS LINES CA<21:0>
2201 010610 047304      AMR210      ;PRINT PATTERN USED FOR AMR<21:0> DATA
2202 010612 000000      0
2203 010614 006301      9$: ASL R1      ;NEXT FLOATING 1 PATTERN
2204 010616 032701 100000      BIT #100000,R1      ;IS ADDRESS PATTERN 40000 DONE?
2205 010622 001735      BEQ 1$      ;NO; CONTINUE
2206 010624 000240      10$: NOP      ;END OF TEST
2207
2208
2209 010626 005237 001464      T71: INC $TESTN      ;UPDATE TEST ID
                               ;*****
                               .SBTTL
                               .SBTTL TES' 71
                               .SBTTL
                               .SBTTL
                               .SBTTL      PA<21:15> FLOATING 1 PATTERN
                               .SBTTL      CA<21:15> FLOATING 1 PATTERN
                               .SBTTL      AMR<21:15> FLOATING 1 PATTERN
                               .SBTTL
                               ;:
                               ;: AFTER EACH FLOATING 1 PAT. CHECK AM BIT READS 1.
                               ;: USE MEMORY MANAGEMENT.
                               ;:*****
2218 010632 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                               ;ERROR/LOOP ON TEST
                               40$: 1$-40$+60000-14      ;TEST START LOCATION
                               0      ;LOOP ON ERROR START LOCATION
                               25$-40$+60000-14      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                               40$: MOV #OFF,CCR      ;LOOP ON ERROR END LOCATION
                               JSR R4,RELCTL      ;DISABLE CACHE
                               10$+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
2219 010660 012737 000002 002054      2$: MOV #2,CMRPAT      ;1ST PATTERN FOR CMR<15:10>
2220 010666 012737 100000 002056      MOV #100000,CHRPAT      ;1ST PATTERN FOR CHR<15:0>
2221
2222 010674 012737 001000 172350      MOV #1000,KPAR4      ;SETUP 1ST PATTRN. FOR PAGE ADDRESS FIELD
2223                               ;KPAR4 CONTAINS THE FLOATING 1
2224                               ;PATTERN AND REPRESENTS THE PAGE ADDRESS FIELD
2225                               ;DATA USED BY MEM. MNGMNT. TO
2226                               ;CONSTRUCT THE PHYSICAL ADDRESS.
2227                               ;1000 IS THE 1ST FLOATING 1 PATTERN
2228                               ;WHICH WILL BE CONSTRUCTED AS PHYS. ADDRESS 100000.
2229 010702 012737 060104 000004      1$: MOV #6$-2$+60000,4      ;ALLOW FOR NEX TRAP

```

```

2230 010710 012737 000340 000004      MOV #340,6
2231 010716 113737 002054 177751      MOVB CMRPAT,CMR+1
2232 010724 013737 002056 177752      MOV CHRPAT,CHR
2233 010732 012737 000001 177572      MOV #1,SRO
2234 010740 012737 000020 172516      MOV #20,SR3
2235 010746 105037 177750      CLRB CMR
2236 010752 005737 100000      TST 100000
2237
2238
2239 010756 000240      NOP
2240 010760 000240      NOP
2241 010762 000401      BR 7$
2242 010764 022626      6$: CMP (SP)+,(SP)+
2243 010766 005037 177572      7$: CLR SRO
2244 010772 005037 172516      CLR SR3
2245
      25$: NOP
      NOP
2246 011002 032737 000010 177750      BIT #AM,CMR
2247 011010 001044      BNE 9$
2248 011012 012737 000006 000004      MOV #6,4
2249 011020 005037 000006      CLR 6
2250 011024 005037 047306      CLR AMR210+2
2251
2252 011030 005037 047302      CLR CA210+2
2253 011034 013737 172350 047304      MOV KPAR4,AMR210
2254 011042 013737 172350 047300      MOV KPAR4,CA210
2255 011050 012737 000011 002052      MOV #9,LOOP
2256 011056 006237 047304      8$: ASR AMR210
2257 011062 006237 047300      ASR CA210
2258 011066 042737 100000 047304      BIC #100000,AMR210
2259 011074 042737 100000 047300      BIC #100000,CA210
2260 011102 005337 002052      DEC LOOP
2261 011106 001363      BNE 5$
2262
2263
      011110 104000      ERROR
      ;ERROR
      ;-----
      011112 011110      .-2
2264
2265 011114 047300      CA210
2266
2267 011116 047304      AMR210
2268 011120 000000      0
2269 011122 006337 172350      9$: ASL KPAR4
2270 011126 103405      BCS 8$
2271 011130 006337 002054      ASL CMRPAT
2272 011134 006337 002056      ASL CHRPAT
2273 011140 000660      BR 1$
2274 011142 012737 000006 000004      8$: MOV #6,4
2275 011150 005037 000006      CLR 6
2276 011154 000240      10$: NOP
2277
2278
2279
2280

```

2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291

.SBTTL
.SBTTL *
.SBTTL *
.SBTTL
.SBTTL *

TAG STORE DATA TESTS- LOADING TAG STORE FROM
ADDRESS MATCH REGISTERS

011156 005237 001464

T72: INC \$TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 72
.SBTTL

2292
2293
2294
2295
2296
2297

:*****
.SBTTL
.SBTTL ALL 0'S TO TAG STORE ADDRESS LOCATION 0000.
.SBTTL

011162 000004

:*****
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40\$;TEST START LOCATION
011164 011174 40\$
011166 070000 1\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
011170 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
011172 070042 25\$-40\$+70000-14 ;LOOP ON ERROR END LOCATION
011174 012737 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE
011202 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
011206 011344 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

2298 011210 005037 177752
2299 011214 112737 000003 177750
2300
2301
2302
2303
2304
2305 011222 012737 000015 177746
2306 011230 005737 040000
2307 011234 005737 060000
2308 011240 005737 060000
2309
2310 011244 013737 177752 047310
2311
011252 000240
011254 000240

1\$: CLR CHR ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
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>
;NO UCB SO AS TO WRITE ENABLE TAG STORE
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
MOV CHR,CHR157
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
NOP ;FOR LOOP ON ERROR

2312 011256 105037 177750
2313 011262 012737 001015 177746
2314 011270 042737 000177 047310
2315 011276 005737 047310
2316 011302 001417
2317 011304 012737 000007 002052
2318 011312 006237 047310

CLR# CMR ;DISABLE MAINTENANCE MODE
MOV #OFF,CCR
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 TYPEOU'
2\$: ASR CHR157

```

2319 011316 042737 100000 017310      BIC #100000,CHR157
2320 011324 005337 002052              DEC LOOP
2321 011330 001370                      BNE 2$
2322                                ERROR                      ;ERROR
                                           ;-----
                                011332 104000
                                011334 011332              .-2
2323                                ;TAG STORE DATA TESTS
2324                                ;READING TAGD<21:13> THRU CHR<15:07>
2325                                ;DID NOT RESULT IN ALL 0'S.
2326 011336 047310                      CHR157
2327 011340 000000                      0
2328 011342 000240                      10$: NOP                      ;END OF TEST
2329
2330
2331 011344 005237 001464      T73: INC $TESTN                      ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 73
                                .SBTTL
2332                                ;*****
2333                                .SBTTL
2334                                .SBTTL ALL 1'S TO TAG STORE ADDRESS LOCATION 0000.
2335                                .SBTTL
2336                                ;*****
2337 011350 000004                      SCPCND                      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                40$                      ;ERROR/LOOP ON TEST
                                1$-40$+70000-14          ;TEST START LOCATION
                                0                          ;LOOP ON ERROR START LOCATION
                                25$-40$+70000-14          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                40$: MOV #OFF,CCR          ;LOOP ON ERROR END LOCATION
                                JSR R4,RELCTM             ;DISABLE CACHE
                                10$+2                    ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST

                                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                ;ARE RE-LOCATED TO HI CACHE SPACE

2338 011376 012737 177777 177752      1$: MOV #-1,CHR                      ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
2339 011404 112737 000003 177750      MOVB #HODO+TDAR,CMR          ;ALLOWS CACHE TAG FIELD BITS TO BE
2340                                ;WRITTEN TO CHR<15:07> ONLY DURING
2341                                ;THE DESTINATION MEMORY ACCESS
2342                                ;OF AN INSTRUCTION
2343                                ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2344                                ;INTO FROM AMR<8:0>
2345 011412 012737 000015 177746      MOV #15,CCR                  ;NO UCB SO AS TO WRITE ENABLE
2346 011420 005737 040000              TST 40000
2347 011424 005737 060000              TST 60000
2348 011430 005737 060000              TST 60000
2349                                ;WRITE INTO TAG STORE
2350 011434 013737 177752 047310      MOV CHR,CHR157              ;WRITE TAG FIELD DATA FROM CACHE ADDRESS
2351                                ;LOCATION 0000 INTO CHR.
                                ;SAVE CHR DATA
                                011442 000240          25$: NOP                      ;INSTRUCTION 'JMP 1$' PLACED HERE
                                011444 000240          NOP                          ;FOR LOOP ON ERROR
2352 011446 105037 177750              CLRB CMR                      ;DISABLE MAINTENANCE MODE
2353 011452 012737 001015 177746      MOV #OFF,CCR

```



```

2392 011634 013737 177752 047310      MOV CHR,CHR157      ;SAVE CHR DATA
2393                                     25$:  NOP            ;INSTRUCTION 'JMP 1$' PLACED HERE
      011642 000240                                     NOP            ;FOR LOOP ON ERROR
      011644 000240

2394 011646 105037 177750      CLRB CMR            ;DISABLE MAINTENANCE MODE
2395 011652 012737 001015 177746      MOV #OFF,CCR       ;DISABLE CACHE
2396 011660 012737 000007 002052      MOV #7,LOOP        ;PREPARE CHR157 FOR COMPARISON
2397 011666 006237 047310      3$:  ASR CHR157
2398 011672 042737 100000 047310      BIC #100000,CHR157
2399 011700 005337 002052      DEC LOOP
2400 011704 001370      BNE 3$
2401 011706 023737 047314 047310      CMP CHR80,CHR157   ;CHECK FOR CORRECT PATTERN
2402 011714 001405      BEQ 9$             ;PASS
2403
      011716 104000      ERROR              ;ERROR
      011720 011716      .-2                ;-----

2404                                     ;TAG STORE DATA TESTS
2405                                     ;READING CHR<15:07> FOR TAGD<21:13>
2406                                     ;DID NOT RESULT IN CORRECT FLOATING
2407                                     ;1 PATTERN.
2408 011722 047310      CHR157             ;PRINT CHR<15:07>
2409 011724 047314      CHR80              ;PRINT FLOATING 1 PATTERN LOADED
2410                                     ;INTO CHR<8:0>
2411 011726 000000      0
2412 011730 006337 047314      9$:  ASL CHR80      ;NEXT PATTERN
2413 011734 032737 001000 047314      BIT #1000,CHR80    ;IF PATTERN 400 DONE;FINISHED
2414 011742 001715      BEQ 1$             ;IF NOT, NEXT PASS
2415 011744 000240      10$: NOP           ;END OF TEST
2416
2417
2418 011746 005237 001464      T75: INC $TESTN    ;UPDATE TEST ID
      ;*****
      .SBTTL
      .SBTTL TEST 75
      .SBTTL
      .SBTTL
2419                                     ;WRITE AND READ 0'S TO ALL LOW CACHE TAG STORE ADDRESS LOCATIONS
2420                                     ;(0000 TO 3777)
2421                                     .SBTTL
2422                                     .SBTTL
2423                                     ;*****
2424 011752 000004      SCPCND            ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      011754 011764      40$              ;ERROR/LOOP ON TEST
      011756 070014      1$-40$+70000-14 ;TEST START LOCATION
      011760 000000      0                  ;LOOP ON ERROR START LOCATION
      011762 070044      25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      011764 012737 001015 177746      40$: MOV #OFF,CCR     ;LOOP ON ERROR END LOCATION
      011772 004437 002352      JSR R4,RELCTH     ;DISABLE CACHE
      011776 012166      10$+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

2425 012000 005037 177752      CLR CHR            ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
2426 012004 012705 060000      MOV #60000,R5     ;ADDRESS 60000 INTO R5
    
```

```

2427 012010 012703 040000          MOV #40000,R3          ;ADDRESS 40000 INTO R3
2428 012014 012737 000015 177746 1$:  MOV #15,CCR           ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2429 012022 112737 000003 177750      MOVB #HODO+TDAR,CMR   ;ALLOWS CACHE TAG FIELD BITS TO BE
2430                                     ;WRITTEN TO CHR<15:07> ONLY DURING
2431                                     ;THE DESTINATION MEMORY ACCESS
2432                                     ;OF AN INSTRUCTION
2433                                     ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2434                                     ;INTO FROM AMR<8:0>
2435 012030 005713          TST (R3)              ;
2436 012032 005715          TST (R5)              ;WRITE INTO TAG STORE
2437 012034 005715          TST (R5)              ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2438                                     ;LOCATION SPECIFIED BY CA<12:1> IN R5.
2439 012036 013737 177752 047310      MOV CHR,CHR157        ;SAVE CHR DATA
2440
2440 012044 000240          25$:  NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
2440 012046 000240          NOP                    ;FOR LOOP ON ERROR
2441 012050 105037 177750          CLR B CMR             ;DISABLE MAINTENANCE MODE
2442 012054 012737 001015 177746      MOV #OFF,CCR          ;DISABLE CACHE
2443 012062 042737 000177 047310      BIC #177,CHR157      ;PREPARE CHR157 FOR ERROR CHECK
2444 012070 005737 047310          TST CHR157           ;BITS <15:07> SHOULD BE ALL 0'S
2445 012074 001424          BEQ 9$               ;PASS
2446 012076 010537 047302          MOV R5,CA210+2       ;SAVE CACHE ADDRESS USED: CA<21:0>
2447 012102 005037 047300          CLR CA210
2448 012106 012737 000007 002052      MOV #7,LOOP           ;ERROR;PREPARE CHR157 FOR TYPEOUT
2449 012114 006237 047310          4$:  ASR CHR157
2450 012120 042737 100000 047310      BIC #100000,CHR157
2451 012126 005337 002052          DEC LOOP
2452 012132 001370          BNE 4$
2453
2453 012134 104000          ERROR                ;ERROR
2453                                     ;-----
2453 012136 012134          .-2
2454                                     ;TAG STORE DATA TESTS
2455                                     ;READING TAGD<21:13> THRU CHR<15:07>
2456                                     ;DID NOT RESULT IN ALL 0'S.
2457 012140 047310          CHR157                ;PRINT CHR<15:07>
2458 012142 047300          CA210                ;PRINT CA<21:0> ADDRESS USED
2459                                     ;BITS <12:1> IS THE CACHE TAG STORE ADDRESS
2460                                     ;LOCATION FAILURE
2461 012144 000000          0
2462 012146 062705 000002          9$:  ADD #2,R5           ;NEXT CACHE STORE LOCATION
2463 012152 062703 000002          ADD #2,R3
2464 012156 020527 070000          CMP R5,#70000        ;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN CHECKED?
2465 012162 001314          BNE 1$                ;NO
2466 012164 000240          10$:  NOP                ;END OF TEST
2467
2468
2469 012166 005237 001464          T76:  INC $TESTN        ;UPDATE TEST ID
2470                                     ;*****
2470 .SBTTL
2471 .SBTTL TEST 76
2472 .SBTTL
2473 .SBTTL WRITE AND READ 1'S TO ALL LOW CACHE TAG STORE ADDRESS LOCATIONS
2473 .SBTTL (0000 TO 3777)
2473 .SBTTL

```

```

2474
2475 012172 000004          :*****
                                SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                40$            ;ERROR/LOOP ON TEST
                                1$-40$+70000-14 ;TEST START LOCATION
                                0             ;LOOP ON ERROR START LOCATION
                                25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
012174 012204          40$: MOV #OFF,CCR ;DISABLE CACHE
012176 070016          JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
012200 000000          10$+2 ;ADDRESS OF START OF NEXT TEST
012202 070046
012204 012737 001015 177746
012212 004437 002352
012216 012412

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

2476 012220 012737 177777 177752      MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
2477 012226 012705 060000      MOV #60000,R5 ;ADDRESS 60000 INTO R5
2478 012232 012703 040000      MOV #40000,R3 ;ADDRESS 40000 INTO R3
2479 012236 012737 000015 177746      1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2480 012244 112737 000003 177750      MOV#B #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2481 ;WRITTEN TO CHR<15:07> ONLY DURING
2482 ;THE DESTINATION MEMORY ACCESS
2483 ;OF AN INSTRUCTION
2484 ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2485 ;INTO FROM AMR<8:0>
2486 012252 005713      TST (R3)
2487 012254 005715      TST (R5) ;WRITE INTO TAG STORE
2488 012256 005715      TST (R5) ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2489 ;LOCATION SPECIFIED BY CA<12:1> IN R5
2490 012260 013737 177752 047310      MOV CHR,CHR157 ;SAVE CHR DATA
2491
012266 000240      25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
012270 000240      NOP ;FOR LOOP ON ERROR

2492 012272 105037 177750      CLR#B CMR ;DISABLE MAINTENANCE
2493 012276 012737 001015 177746      MOV #OFF,CCR
2494 012304 042737 000177 047310      BIC #177,CHR157
2495 012312 022737 177600 047310      CMP #177600,CHR157 ;BITS <15:07> SHOULD BE ALL 1'S
2496 012320 001424      BEQ 9$ ;PASS
2497 012322 010537 047302      MOV R5,CA210+2 ;SAVE CACHE ADDRESS USED: CA<21:0>
2498 012326 005037 047300      CLR CA210
2499 012332 012737 000007 002052      MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
2500 012340 006237 047310      4$: ASR CHR157
2501 012344 042737 100000 047310      BIC #100000,CHR157
2502 012352 005337 002052      DEC LOOP
2503 012356 001370      BNE 4$
2504
012360 104000      ERROR ;ERROR
012362 012360      .-2 ;-----

2505 ;TAG STORE DATA TESTS
2506 ;READING TAGD<21:13> THRU CHR<15:07>
2507 ;DID NOT RESULT IN ALL 1'S.
2508 012364 047300      CA210 ;PRINT CACHE ADDRESS CA<21:0>
2509 012366 047310      CHR157 ;PRINT CHR<15:07>
2510 012370 000000      0
2511 012372 062705 000002      9$: ADD #2,R5 ;NEXT CACH LOCATION
2512 012376 062703 000002      ADD #2,R3
    
```



```

2513 012402 020527 070000      CMP R5,#70000      ;HAVE ALL LOCATIONS BEEN DONE?
2514 012406 001313              BNE 1$            ;NO
2515 012410 000240              10$: NOP          ;END OF TEST
2516
2517
2518
2519 012412 005237 001464      77: INC $TESTN    ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 77
.SBTTL
2520 .SBTTL
2521 .SBTTL WRITE AND READ 0'S TO ALL HI CACHE TAG STORE ADDRESS LOCATIONS
2522 .SBTTL (4000 TO 7777)
2523 .SBTTL
2524 ;*****
2525 012416 000004              SCPCND           ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
012420 012430              40$            ;LOOP ON ERROR START LOCATION
012422 060014              1$-40$+60000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
012424 000000              0               ;LOOP ON ERROR END LOCATION
012426 060044              25$-40$+60000-14 ;DISABLE CACHE
012430 012737 001015 177746    40$: MOV #OFF,CCR  ;LOCATE TEST CODE TO LOW CACHE SPACE
012436 004437 002324          JSR R4,RELCTL   ;ADDRESS OF START OF NEXT TEST
012442 012632

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

2526 012444 005037 177752      CLR CHR          ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
2527 012450 012705 070000      MOV #70000,R5   ;ADDRESS 70000 INTO R5
2528 012454 012703 050000      MOV #50000,R3   ;ADDRESS 50000 INTO R3
2529 012460 012737 000015 177746 1$: MOV #15,CCR    ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2530 012466 112737 000003 177750 MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2531                                     ;WRITTEN TO CHR<15:07> ONLY DURING
2532                                     ;THE DESTINATION MEMORY ACCESS
2533                                     ;OF AN INSTRUCTION
2534                                     ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2535                                     ;INTO FROM AMR<8:0>
2536 012474 005713              TST (R3)
2537 012476 005715              TST (R5)
2538 012500 005715              TST (R5)
2539                                     ;WRITE INTO TAG STORE
2540 012502 013737 177752 047310 MOV CHR,CHR157  ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2541                                     ;LOCATION SPECIFIED BY CA<12:1> IN R5.
2541                                     ;SAVE CHR DATA
012510 000240              25$: NOP        ;INSTRUCTION 'JMP 1$' PLACED HERE
012512 000240              NOP            ;FOR LOOP ON ERROR

2542 012514 105037 177750      CLRB CMR        ;DISABLE MAINTENANCE MODE
2543 012520 012737 001015 177746 MOV #OFF,CCR    ;DISABLE CACHE
2544 012526 042737 000177 047310 BIC #177,CHR157 ;PREPARE CHR157 FOR ERROR CHECK
2545 012534 005737 047310      TST CHR157     ;BITS <15:07> SHOULD BE ALL 0'S
2546 012540 001424              BEQ 9$         ;PASS
2547 012542 010537 047302      MOV R5,CA210+2 ;SAVE CACHE ADDRESS USED: CA<21:0>
2548 012546 005037 047300      CLR CA210
2549 012552 012737 000007 002052 MOV #7,LOOP
2550 012560 006237 047310      4$: ASR CHR157 ;ERROR;PREPARE CHR157 FOR *YPEOUT
    
```

2551	012564	042737	100000	047310
2552	012572	005337	002052	
2553	012576	001370		

BIC #100000,CHR157
 DEC LOOP
 BNE 48

2554
 012600 104000

ERROR

;ERROR
 ;-----

012602 012600

.-2

2555
 2556
 2557

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

2559	012604	047310	
2560	012606	047300	
2561			
2562			
2563	012610	000000	
2564	012612	062705	000002
2565	012616	062703	000002
2566	012622	020527	100000
2567	012626	001314	

CHR157
CA210

:PRINT CHR<15:07>
:PRINT CA<21:0> ADDRESS USED
:BITS <12:1> IS THE CACHE TAG STORE ADDRESS
:LOCATION FAILURE

```

0
9$: ADD #2,R5
    ADD #2,R3
    CMP R5,#100000
    BNE 1$

```

:NEXT CACHE STORE LOCATION
:HAVE ALL HI CACHE ADDRESS LOCATIONS BEEN CHECKED?
:NO

```

2569 012630 000240          10$:  NOP                ;END OF TEST
2570
2571
2572 012632 005237 001464 1100:  INC $TESTN          ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 100
.SBTTL
.SBTTL
.SBTTL WRITE AND READ 1'S TO ALL HI CACHE TAG STORE ADDRESS LOCATIONS
.SBTTL (4000 TO 7777)
.SBTTL
;*****
2573
2574
2575
2576
2577
2578 012636 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                40$                ;TEST START LOCATION
                                1$-40$+60000-14      ;LOOP ON ERROR START LOCATION
                                0                    ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$-40$+60000-14      ;LOOP ON ERROR END LOCATION
                                40$:  MOV #OFF,CCR          ;DISABLE CACHE
                                JSR R4,RELCTL         ;LOCATE TEST CODE TO LOW CACHE SPACE
                                10$+2              ;ADDRESS OF START OF NEXT TEST

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

2579 012664 012737 177777 177752          MOV #-1,CHR          ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
2580 012672 012705 070000          MOV #70000,R5        ;ADDRESS 70000 INTO R5
2581 012676 012703 050000          MOV #50000,R3        ;ADDRESS 50000 INTO R3
2582 012702 012737 000015 177746 1$:  MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2583 012710 112737 000003 177750          MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2584                                     ;WRITTEN TO CHR<15:07> ONLY DURING
2585                                     ;THE DESTINATION MEMORY ACCESS
2586                                     ;OF AN INSTRUCTION
2587                                     ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2588                                     ;INTO FROM AMR<8:0>
2589 012716 005713          TST (R3)
2590 012720 005715          TST (R5)
2591 012722 005715          TST (R5)
2592                                     ;WRITE INTO TAG STORE
2593 012724 013737 177752 047310          MOV CHR,CHR157      ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2594                                     ;LOCATION SPECIFIED BY CA<12:1> IN R5
2595                                     ;SAVE CHR DATA
                                25$:  NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP                ;FOR LOOP ON ERROR

2596 012736 105037 177750          CLR B CMR           ;DISABLE MAINTENANCE
2597 012742 012737 001015 177746          MOV #OFF,CCR
2598 012750 042737 000177 047310          BIC #177,CHR157
2599 012756 022737 177600 047310          CMP #177600,CHR157 ;BITS <15:07> SHOULD BE ALL 1'S
2600 012764 001424          BEQ 9$             ;PASS
2601 012766 010537 047302          MOV R5,CA210+2     ;SAVE CACHE ADDRESS USED: CA<21:0>
2602 012772 005037 047300          CLR CA210
2603 012776 012737 000007 002052          MOV #7,LOOP        ;ERROR;PREPARE CHR157 FOR TYPEOUT
2604 013004 006237 047310          4$:  ASR CHR157
2605 013010 042737 100000 047310          BIC #100000,CHR157
2606 013016 005337 002052          DEC LOOP
                                BNE 4$

```

```

2607      013024  104000          ERROR          ;ERROR
          013026  013024          .-2          ;-----
2608
2609
2610
2611 013030  047300          CA210      ;TAG STORE DATA TESTS
2612 013032  047310          CHR157     ;READING TAGD<21:13> THRU CHR<15:07>
2613 013034  000000          0          ;DID NOT RESULT IN ALL 1'S.
2614 013036  062705  000002  9$:      ADD #2,R5   ;PRINT CACHE ADDRESS CA<21:0>
2615 013042  062703  000002          ADD #2,R3   ;PRINT CHR<15:07>
2616 013046  020527  100000          CMP R5,#100000 ;NEXT CACH LOCATION
2617 013052  001313          BNE 1$     ;HAVE ALL LOCATIONS BEEN DONE?
2618 013054  000240          10$:     NOP      ;NO
          ;END OF TEST
2619
2620
2621
2622
2623      .SBTTL *
2624      .SBTTL *
2625      .SBTTL *
2626      .SBTTL *
2627      .SBTTL *
2628      .SBTTL *
2629
2630
2631
2632
2633
2634 013056  005237  001464  T101:      INC $TESTN    ;UPDATE TEST ID
          ;*****
          .SBTTL
          .SBTTL TEST 101
          .SBTTL
          .SBTTL
2635      .SBTTL CHECK LOADING OF TAG STORE DATA(TAG WRTD<21:13>) FROM
2636      .SBTTL CACHE ADDRESS LINES CA<21:13>.
2637      .SBTTL
2638      .SBTTL WRITE ALL 0'S IN TAG STORE LOCATION 0000 FROM CA<21:13>
2639      .SBTTL
2640      .SBTTL
2641      ;*****
2642 013062  000004          SPCOND     ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          013064  013074          40$        ;ERROR/LOOP ON TEST
          013066  070000          1$-40$+70000-14 ;TEST START LOCATION
          013070  000000          0          ;LOOP ON ERROR START LOCATION
          013072  070036          25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          013074  012737  001015  177746  40$:     MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
          013102  004437  002352          JSR R4,RELCTH ;DISABLE CACHE
          013106  013240          10$+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
2643 013110  112737  000002  177750  1$:      MOVB #HODG,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2644
          ;WRITTEN TO CHR<15:07> ONLY DURING

```

```

2645                                     ;THE DESTINATION MEMORY ACCESS
2646                                     ;OF AN INSTRUCTION
2647 013116 012737 000015 177746      MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2648 013124 005737 040000              TST 40000
2649 013130 005737 000000              TST 0                ;WRITE ALL 0'S INTO TAG STORE LOCATION 0000
2650                                     ;FROM CACHE ADDRESS CA<21:13>
2651 013134 005737 000000              TST 0                ;WRITE TAG STORE DATA FROM LOCATION
2652                                     ;0000 INTO CHR<15:07>.
2653 013140 013737 177752 047310      MOV CHR,CHR157       ;SAVE CHR DATA
2654
    013146 000240                25$: NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
    013150 000240                NOP                    ;FOR LOOP ON ERROR

2655 013152 105037 177750              CLR B CMR            ;DISABLE MAINTENANCE MODE
2656 013156 012737 001015 177746      MOV #OFF,CCR         ;DISABLE CACHE
2657 013164 042737 000177 047310      BIC #177,CHR157     ;PREPARE CHR157 FOR ERROR CHECK
2658 013172 005737 047310              TST CHR157          ;BITS <15:07> SHOULD BE ALL 0'S
2659 013176 001417                    BEQ 10$              ;PASS
2660 013200 012737 000007 002052      MOV #7,LOOP         ;ERROR;PREPARE CHR157 FOR TYPEOUT
2661 013206 006237 047310                2$: ASR CHR157
2662 013212 042737 100000 047310      BIC #100000,CHR157
2663 013220 005337 002052              DEC LOOP
2664 013224 001370                BNE 2$
2665
    013226 104000                ERROR                    ;ERROR
    013230 013226                .-2                        ;-----

2666                                     ;TAG STORE DATA TESTS
2667                                     ;READING TAGD<21:13> THRU CHR<15:07>
2668                                     ;DID NOT RESULT IN ALL 0'S.
2669 013232 047310                CHR157                ;PRINT CHR<15:07>
2670 013234 000000                0
2671 013236 000240                10$: NOP                ;END OF TEST
2672
2673
2674
2675
2676
2677
2678
2679
2680 013240 005237 001464      T102: INC $TESTN        ;UPDATE TEST ID
    ;*****
    ;SBTTL
    ;SBTTL TEST 102
    ;SBTTL
    ;SBTTL
    ;SBTTL WRITE FLOATING 1 ACROSS 0'S INTO TAG STORE LOCATION 0000
    ;SBTTL FROM CA<21:13> USING AVAILABLE MEMORY.
    ;SBTTL PROCEDURE: STARTING AT 8K BOUNDARY(ADDR. 20000) CHECK
    ;SBTTL FOR AVAILABLE FLOATING ADDRESS UP TO ADDR. 10000000
    ;SBTTL WHEN THE FLOATING ADDRESS EXISTS PERFORM THE TEST.
    ;*****
2681                                     SPCOND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
2682                                     ;ERROR/LOOP ON TEST
2683                                     40$                    ;TEST START LOCATION
2684                                     1$-40$+70000-14       ;LOOP ON ERROR START LOCATION
2685
2686
2687
2688 013244 000004
    013246 013256
    013250 070006

```

```

013252 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
013254 070100          25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
013256 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
013264 004437 002352      JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
013270 013556          10$+2      ;ADDRESS OF START OF NEXT TEST

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

2689 013272 012737 000200 172350 2$: MOV #200,KPAR4 ;KPAR4 CONTAINS THE FLOATING 1 PATTERN
2690                                     ;AND REPRESENTS THE THE PAGE ADDRESS FIELD
2691                                     ;DATA USED BY MEMORY MNGMNT. TO CONSTRUCT
2692                                     ;THE PHYSICAL ADDRESS. 200 IS THE 1ST
2693                                     ;FLOATING 1 PATTERN WHICH WILL BE CONSTRUCTED
2694                                     ;TO ADDRESS 20000.
2695 013300 012737 070240 000004 1$: MOV #7$-2$+70000,4 ;ALLOW FOR NEX TRAP
2696 013306 012737 000340 000006      MOV #340,6
2697 013314 112737 000002 177750      MOVB #HODO,CMR ;ALLOWS CACHE TAG STORE TO BE WRITTEN
2698                                     ;TO CHR<15:07> ONLY DURING THE DESTINATION
2699                                     ;MEMORY ACCESS OF AN INSTRUCTION.
2700 013322 012737 000001 177572      MOV #1,SRO ;ENABLE MEMORY MNGMNT.
2701 013330 012737 000020 172514      MOV #20,SR3 ;ENABLE 22-BIT MAPPING
2702 013336 012737 000015 177746      MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2703 013344 005737 040000          TST 40000 ;
2704 013350 005737 100000          TST 100000 ;CHOOSES KPAR4 FOR ADDRESSING. PHYSICAL
2705                                     ;ADDRESS WILL BE DETERMINED BY FLOATING
2706                                     ;PATTERN USED IN KPAR4. TAG STORE WILL
2707                                     ;BE WRITTEN WITH DATA PLACED ON CA<21:13> ADDRESS LI
2708 013354 000240          NOP
2709 013356 000240          NOP ;NO TRAP
2710 013360 005737 100000          TST 100000 ;WRITE TAG STORE DATA FROM LOCATION
2711                                     ;0000 INTO CHR<15:07>.
2712 013364 013737 177752 047310      MOV CHR,CHR157 ;SAVE CHR INFO.
2713                                     ;
013372 000240          25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
013374 000240          NOP ;FOR LOOP ON ERROR

2714 013376 005037 177572          CLR SRO ;DISABLE MEM MNGMENT.
2715 013402 005037 172516          CLR SR3
2716 013406 012737 001015 177746      MOV #OFF,CCR ;DISABLE CACHE
2717 013414 105037 177750          CLRB CMR ;DISABLE MAINTENANCE
2718 013420 042737 000177 047310      BIC #177,CHR157
2719 013426 023737 172350 047310      CMP KPAR4,CHR157 ;IS THERE ERROR?
2720 013434 001437          BEQ 9$ ;PASS
2721 013436 012737 000006 000004      MOV #6,4
2722 013444 005037 000006          CLR 6
2723 013450 013737 172350 047312      MOV KPAR4,CA2113 ;SAVE PATTERN USED FOR CA<21:13>.
2724 013456 012737 000007 002052      MOV #7,LOOP ;PREPARE CHR157 AND CA2113 FOR ERROR PRINT
2725 013464 006237 047310          5$: ASR CHR157
2726 013470 006237 047312          ASR CA2113
2727 013474 042737 100000 047310      BIC #100000,CHR157
2728 013502 042737 100000 047312      BIC #100000,CA2113
2729 013510 005337 002052          DEC LOOP
2730 013514 001363          BNE 5$
2731 013516 104000          ERROR ;ERROR
                                     ;-----

```

```

013520 013516      .-2
2732                                     ;TAG STORE TESTS
2733                                     ;READING CHR<15:07> FOR TAG DATA (TAGD <21:13>)
2734                                     ;DID NOT RESULT IN CORRECT ADDRESS PATTERN
2735                                     ;LOADED FROM CA<21:13>.
2736 013522 047310      CHR157                                     ;PRINT CHR<15:07>
2737 013524 047312      CA2113                                     ;PRINT ADDRESS PATTERN USED: CA<21:13>
2738 013526 000000      0                                         ;PRINT TERMINATE
2739 013530 000401      BR 9$                                       ;NEXT PATTERN
2740 013532 022626      7$: CMP (SP)+,(SP)+                          ;RESTORE STACK DUE TO INTERRUPT
2741 013534 006337 172350 9$: ASL KPAR4                               ;NEXT PATTERN;IF PHYSICAL ADDRESS
2742                                     ;10000000 HAS BEEN DONE; FINISHED
2743 013540 103257      BCC 1$                                       ;RESTORE VECTORS
2744 013542 012737 000006 000004      MOV #6,4
2745 013550 005037 000006      CLR 6
2746 013554 000240      10$: NOP                                     ;END OF TEST
2747
2748
2749
2750
2751 013556 005237 001464      T103: INC $TESTN                                     ;UPDATE TEST ID
2752                                     ;*****
2753                                     .SBTTL
2754                                     .SBTTL TEST 103
2755                                     .SBTTL
2756                                     .SBTTL
2757                                     .SBTTL WRITE FLOATING 1 ACROSS 0'S INTO TAG STORE ADDRESS LOCATION 0000
2758                                     .SBTTL FROM CA<21:13> USING RMI REGISTER (G5179)
2759                                     ;:
2760                                     ;: PROCEDURE: START AT 16K BOUNDARY (ADDR. 100000) AND CHECK FOR
2761                                     ;: AVAILABLE FLOATING ADDRESSES UP TO ADDR. 1000000
2762                                     ;: WHEREEVER A FLOATING ADDRESS DOES NOT EXIST USE
2763                                     ;: THE RMI REGISTER. IF ADDRESS EXISTS DO NOT PERFORM
2764                                     ;: THE TEST SINCE THAT LOCATION WOULD HAVE BEEN TESTED
2765                                     ;: BY THE PREVIOUS TEST.
2766                                     ;:
2767                                     ;:*****
2768 013562 000004      SCPCND                                     ;SCOPE CONDITIONS:GO SET UP FOR LOUP ON
2769                                     ;:ERROR/LOOP ON TEST
2770                                     ;:TEST START LOCATION
2771 013564 013574      40$                                     ;LOOP ON ERROR START LOCATION
2772 013556 070050      1$-40$+'0000-14                             ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
2773 013570 000000      0                                         ;LOOP ON ERROR END LOCATION
2774 013572 070166      25$-40$+70000-14                             ;DISABLE CACHE
2775 013574 012737 001015 177746 40$: MOV #0,F,CCR                ;LOCATE TEST CODE TO HIGH CACHE SPACE
2776 013602 004437 002352      JSR R4,RELCTH                             ;ADDRESS OF START OF NEXT TEST
2777 013606 014334      10$+2
2778                                     ;THE FOLLOWING LOCATIONS INCLUDING 10$
2779                                     ;:ARE RELOCATED TO HI CACHE SPACE
2780
2781 2764 013610 132737 000200 001501      2$: BITB #APTSIZE,$ENVM                                     ;DOES APT SIZE?
2782 2765 013616 001405      BEQ 11$                                       ;NO GO AUTOSIZE
2783 2766 013620 032737 000200 001504      BIT #200,$USWR                                           ;DOES APT INDICATE
2784 2767                                     ;:THAT RMI REGISTER IS PRESENT
2785 2768 013626 001006      BNE 5$                                       ;YES,USE IT
2786 2769 013630 000555      BR 4$                                       ;APT SAYS DO NOT PERFORM TEST
2787 2770
2788 2771 013632 012737 070352 000004      11$: MOV #3$-2$+70000,4                                     ;AUTO-SIZE FOR RMI,PREPARE FOR TRAP
  
```



```
2772 013640 005737 177770          TST @#177770          ;READ RMI
2773
2774 013644 012737 001000 172350    5$: MOV #1000,KPAR4    ;SETUP MEM. MNG. PAGE 4 FOR FIRST FLOATING
2775                                     ;ADDRESS 100000
2776 013652 012737 000002 047332    MOV #2,FLTPAT        ;SETUP 1ST FLOATING PATTERN FOR RMI
2777                                     ;REG. CORRESPONDING TO ADDRESS 40000
2778 013660 012737 070100 000004    1$: MOV #8$-2$+70000,4 ;SETUP FOR NEX MEMORY
2779 013666 012737 000001 177572    MOV #1,SRO           ;ENABLE MEM.MNGMENT.
2780 013674 012737 000020 172516    MOV #20,SR3          ;ENABLE 22 BIT MAPPING
2781
2782 013702 005737 100000          TST 100000           ;SELECT PAGE 4. READ ADDRESS SPECIFIED
2783                                     ;BY KPAR4.
2784 013706 000512          BR 9$                ;NO TRAP MEMORY LOCATION EXISTS,SO DON'T
2785                                     ;BOTHER TESTING WITH RMI FOR THIS LOCATION
2786 013710 022626          8$: CMP (SP)+,(SP)+  ;TRAP HERE WHEN FLOATING ADDRESS
2787                                     ;LOCATION DOES NOT EXIST.USE RMI FOR TESTING
2788 013712 013701 047332          MOV FLTPAT,R1        ;PREPARE FLTPAT FOR LOADING INTO RMI
2789 013716 005101          COM R1
2790
2791 013720 110137 177770          MOV#B R1,@#177770    ;LOAD RMI REGISTER
2792 013724 112737 000002 177750    MOV#B #HODO,CMR      ;ALLOWS CACHE TAG STORE TO BE WRITTEN
2793                                     ;TO CHR<15:07> DURING THE DESTINATION
2794                                     ;MEMORY ACCESS OF AN INSTRUCTION ONLY
2795 013732 012737 000015 177746    MOV #15,CCR          ;NO UCB TO ENABLE TAG STORE WRITING
2796 013740 052737 000400 177770    BIS #400,@#177770   ;ENABLE RMI
2797 013746 005737 040000          TST 40000
2798 013752 005737 100000          TST 100000           ;SELECT PAGE 4 AND READ FLOATING ADDRESS
2799                                     ;SPECIFIED BY KPAR4. RMI WILL RESPOND
2800                                     ;RESULTING IN THE TAG STORE BEING LOADED
2801                                     ;FROM CA<21:13>
2802 013756 005737 100000          TST 100000           ;WRITE TAG STORE DATA TO CHR<15:07>
2803 013762 013737 177752 047310    MOV CHR,CHR157       ;SAVE CHR DATA
2804 013770 042737 000400 177770    BIC #400,@#177770   ;DISABLE RMI
2805
2806 013776 000240          25$: NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
2807 014000 000240          NOP                  ;FOR LOOP ON ERROR
2808
2806 014002 005037 177572          CLR SRO              ;DISABLE MEM MNGMENT.
2807 014006 005037 172516          CLR SR3
2808 014012 012737 001015 177746    MOV #OFF,CCR         ;DISABLE CACHE
2809 014020 105037 177750          CLR#B CMR           ;DISABLE MAINTENANCE
2810 014024 042737 000177 047310    BIC #177,CHR157
2811 014032 023737 172350 047310    CMP KPAR4,CHR157    ;IS THERE ERROR?
2812 014040 001435          BEQ 9$              ;PASS
2813 014042 012737 000006 000004    MOV #6,4
2814 014050 005037 000006          CLR 6
2815 014054 013737 172350 047312    MOV KPAR4,CA2115    ;SAVE PATTERN USED FOR CA<21:13>.
2816 014062 012737 000007 002052    MOV #7,LOOP         ;PREPARE CHR157 AND CA2113 FOR ERROR PRINT
2817 014070 006237 047310          6$: ASR CHR157
2818 014074 006237 047312          ASR CA2113
2819 014100 042737 100000 047310    BIC #10000J,CHR157
2820 014106 042737 100000 047312    BIC #100000,CA2113
2821 014114 005337 002052          DEC LOOP
2822 014120 001363          BNE 6$
2823
2823 014122 104000          ERROR                ;ERROR
                                     ;-----
```

```

014124 014122      .-2
2824      :TAG STORE TESTS USING RMI REGISTER
2825      :READING CHR<15:07> FOR TAG DATA (TAGD <21:13>)
2826      :DID NOT RESULT IN CORRECT ADDRESS PATTERN
2827      :LOADED FROM CA<21:13>.
2828 014126 047310  CHR157
2829 014130 047312  CA2113
2830 014132 000000  0
2831 014134 006337 172350 9$: ASL KPAR4
2832 014140 006337 047332  ASL FLTPAT
2833 014144 103245  BCC 1$
2834
2835 014146 012737 000006 000004  MOV #6,@#4
2836 014154 005037 000006  CLR @#6
2837 014160 000464  BR 10$
2838
2839 014162 022626 3$: CMP (SP)+,(SP)+
2840 014164 012737 000006 000004 4$: MOV #6,@#4
2841 014172 005037 000006  CLR @#6
2842 014176 005737 001466  TST $PASS
2843 014202 001053  BNE 10$
2844 014204 023737 000042 000046  CMP @#42,@#46
2845 014212 001447  BEQ 10$
2846 014214 104401 014222  TYPE ,65$
014220 000427  BR 64$
2847 014300 104401 014306 64$: .ASCIZ <CRLF>/RMI REGISTER (G5179) NOT USED-SKIP HI ORDER/
014304 000412  TYPE ,67$
66$: .ASCIZ / BIT ADDRESS TEST/<CRLF>
2848
2849 014332 000240 10$: NOP
2850
2851
2852
2853
2854
2855
2856 .SBTTL *
2857 .SBTTL
2858 .SBTTL *
2859 .SBTTL
2860 .SBTTL
2861 .SBTTL *
2862
2863
2864
2865
2866
2867
2868
2869 014334 005237 001464 T104: INC $TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 104
.SBTTL
    
```

TAG STORE ADDRESS LINE TESTS

```

2870 .SBTTL
2871 .SBTTL VERIFY TAG STORE ADDRESS LINES (CA<12:1>)
2872 .SBTTL
2873 .:
2874 .:
2875 .:
2876 .:
2877 .:
2878 .:
2879 .:
2880 .:
2881 .:
2882 .:

```

PROCEDURE: WRITE 0 INTO TAGG PARITY STORE ADDRESS LOCATION 0000.
WRITE BIT PATTERN 00000011 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.

```

2883 014340 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
014342 014352 40$ ;TEST START LOCATION
014344 070032 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
014346 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
014350 070072 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
014352 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
014360 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
014364 014546 10$+2 ;ADDRESS OF START OF NEXT TEST

```

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

```

2884 014366 012737 000002 047332 2$: MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
2885 014374 012702 040000 MOV #40000,R2 ;
2886 014400 012703 060000 MOV #60000,R3 ;
2887 014404 063702 047332 ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
2888 014410 063703 047332 ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
2889 014414 012713 177777 MOV #-1,(R3) ;ALL 1'S TO MAIN MEM. LOCATION
2890 ;SPECIFIED BY R3
2891 014420 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS TAG STORE BIT TO BE
2892 ;WRITTEN TO CHR<15:07> ONLY DURING THE
2893 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2894 014426 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2895 014434 005737 040000 TST 40000
2896 014440 005737 000000 TST 0 ;READ UPDATE; WRITE ALL 0'S INTO CACHE
2897 ;TAG STORE LOCATION 0000.
2898 014444 005712 TST (R2)
2899 014446 005713 TST (R3) ;READ UPDATE;WRITE BIT PATTERN 00000011
2900 ;INTO TAG STORE LOCATION SPECIFIED
2901 ;BY R3'S BITS 1 THRU 12: CA<12:1>
2902 014450 005737 060000 TST 60000 ;LOAD TAG STORE LOCATION 0000 INTO CHR
2903 014454 013701 177752 MOV CHR,R1 ;SAVE CHR CONTENTS
2904 014460 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
014462 000240 NOP ;FOR LOOP ON ERROR
2905 014464 105037 177750 CLR B CMR ;DISABLE MAINT. MODE
2906 014470 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
2907 014476 042701 000177 BIC #177,R1 ;INTERESTED IN ONLY BITS 15:07
2908 014502 005701 TST R1 ;CHECK FOR ALL 0'S
2909 014504 001411 BEQ 9$ ;PASS
2910 014506 013737 047332 047322 MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
2911 014514 006237 047322 ASR CA121 ;PREPARE CA121 FOR TYPEOUT

```

```

2912 014520 104000 ERROR ;ERROR
;-----
014522 014520 .-2
2913 ;TAG STORE ADDRESS LINE TESTS
2914 ;READING CHR<15:07> FOR CACHE TAG STORE
2915 ;DID NOT RESULT IN ALL 0'S
2916 014524 047322 CA121 ;PRINT CACHE TAG STORE ADDRESS LOCATION
2917 ;USED. CA<12:1>. NOTE THAT THE 1 IN
2918 ;THIS PATTERN WILL POINT TO THE ADDRESS
2919 ;LINE THAT POSSIBLY CAUSES ERROR.
2920 014526 000000 0
2921 ;:::
2922 014530 006337 047332 9$: ASL FLTPAT ;NEXT PATTERN
2923 014534 022737 020000 047332 CMP #20000,FLTPAT ;HAS CACHE TAG STORE LOCAT. 4000 BEEN DONE?
2924 014542 001314 BNE 2$ ;NO
2925 014544 000240 . 10$: NOP ;END OF TEST
2926
2927
2928
2929
2930 .SBTTL *
2931 .SBTTL
2932 .SBTTL * DATA STORE TESTS
2933 .SBTTL
2934 .SBTTL *
2935
2936
2937
2938
2939
2940
2941 014546 005237 001464 T105: INC $TESTN ;UPDATE TEST ID
;:*****
;SBTTL
;SBTTL TEST 105
;SBTTL
;:*****
;SBTTL
;SBTTL WRITE ALL 0'S INTO DATA STORE LOCATION 0000.
;SBTTL READ ALL 0'S FROM CACHE DATA REGISTER.
;SBTTL
;:*****
2942
2943
2944
2945
2946
2947
2948 014552 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
014554 014564 40$ ;ERROR/LOOP ON TEST
014556 070004 1$-40$+70000-14 ;TEST START LOCATION
014560 000000 0 ;LOOP ON ERROR START LOCATION
014562 070042 25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
014564 012737 001015 177746 40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
014572 004437 002352 JSR R4,RELCTH ;DISABLE CACHE
014576 014700 10$+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

2949 014600 005037 060000 CLR 60000 ;0'S TO MAIN MEMORY LOCATION

```

```

2950 014604 112737 000002 177750      1$:  MOVB #HODO,CMR      ;ALLOWS CACHE DATA STORE BITS TO BE
2951                                     ;WRITTEN TO CDR<15:0> ONLY DURING THE
2952                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2953 014612 012737 000015 177746      MOV #15,CCR      ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2954 014620 005737 040000      TST 40000
2955 014624 005737 060000      TST 60000
2956                                     ;WRITE ALL 0'S TO DATA STORE
2957                                     ;LOCATION 0000 FROM MAIN MEMORY
2958 014630 005737 060000      TST 60000      ;LOC. 60000
2959                                     ;WRITE DATA STORE BITS FROM
2960 014634 013737 177754 047316      MOV CDR,CDR150  ;LOC. 0000 INTO CDR<15:0>.
2961                                     ;SAVE CDR CONTENTS
      014642 000240      25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
      014644 000240      NOP          ;FOR LOOP ON ERROR
2962 014646 105037 177750      CLRB CMR        ;DISABLE MAINTENANCE
2963 014652 012737 001015 177746      MOV #OFF,CCR    ;DISABLE CACHE
2964 014660 005737 047316      TST CDR15C     ;CHECK FOR 0
2965 014664 001404      BEQ 10$        ;PASS
2966      014666 104000      ERROR          ;ERROR
      014670 014666      .-2          ;-----
2967                                     ; DATA STORE TESTS
2968                                     ;READING CDR<15:0> DID NOT RESULT
2969                                     ;IN ALL 0'S
2970 014672 047316      CDR150         ;PRINT CDR<15:0> DATA READ.
2971 014674 000000      0
2972      :::
2973 014676 000240      10$:  NOP          ;END OF TEST
2974
2975
2976 014700 005237 001464      T106:          INC $TESTN      ;UPDATE TEST ID
      ;*****
      ;SBTTL
      ;SBTTL TEST 106
      ;SBTTL
      ;*****
2977      ;SBTTL
2978      ;SBTTL WRITE ALL 1'S INTO DATA STORE LOCATION 0000.
2979      ;READ ALL 1'S FROM CACHE DATA REGISTER.
2980      ;SBTTL
2981      ;*****
2982      ;*****
2983 014704 000004      SCPCND        ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      014706 014716      40$          ;ERROR/LOOP ON TEST
      014710 070006      1$-40$+70000-14 ;TEST START LOCATION
      014712 000000      0            ;LOOP ON ERROR START LOCATION
      014714 070044      25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      014716 012737 001015 177746      40$:  MOV #OFF,CCR    ;LOOP ON ERROR END LOCATION
      014724 004437 002352      JSR R4,RELCTH  ;DISABLE CACHE
      014730 015036      10$+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
2984 014732 012737 177777 060000      MOV #-1,60000  ;1'S TO MAIN MEMORY LOCATION

```

```

2985 014740 112737 000002 177750      1$:  MOV B #HODO,CMR      ;ALLOWS CACHE DATA STORE BITS TO BE
2986                                     ;WRITTEN TO CDR<15:07> ONLY DURING THE
2987                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2988 014746 012737 000015 177746      MOV #15,CCR      ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2989 014754 005737 040000      TST 40000
2990 014760 005737 060000      TST 60000
2991                                     ;WRITE ALL 1'S TO DATA STORE
2992                                     ;LOCATION 0000 FROM MAIN MEMORY
2993 014764 005737 060000      TST 60000      ;LOC. 60000
2994                                     ;WRITE DATA STORE BITS FROM
2995 014770 013737 177754 047316      MOV CDR,CDR150 ;LOC. 0000 INTO CDR<15:07>.
2996                                     ;SAVE CDR CONTENTS
014776 000240      25$:  NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
015000 000240      NOP      ;FOR LOOP ON ERROR

2997 015002 105037 177750      CLR B CMR      ;DISABLE MAINTENANCE
2998 015006 012737 001015 177746      MOV #OFF,CCR   ;DISABLE CACHE
2999 015014 022737 177777 047316      CMP #-1,CDR150 ;CHECK ALL 1'S
3000 015022 001404      BEQ 10$      ;PASS
3001 015024 104000      ERROR      ;ERROR
015026 015024      .-2      ;-----

3002                                     ; DATA STORE TESTS
3003                                     ;READING CDR<15:0> DID NOT RESULT
3004                                     ;IN ALL 1'S
3005 015030 047316      CDR150      ;PRINT CDR<15:0> DATA READ.
3006 015032 000000      0
3007                                     :::
3008 015034 000240      10$:  NOP      ;END OF TEST
3009
3010
3011 015036 005237 001464      T107:      INC $TESTN      ;UPDATE TEST ID
;*****
;SBTTL
;SBTTL .TEST 107
;SBTTL
;*****
;SBTTL
;SBTTL WRITE FLOATING 1 PATTERN INTO DATA STORE LOCATION 0000.
;SBTTL READ FLOATING 1 PATTERN FROM CACHE DATA REGISTER.
;SBTTL
;*****
3012
3013
3014
3015
3016
3017
3018 015042 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
015044 015054      40$      ;ERROR/LOOP ON TEST
015046 070014      1$-40$+70000-14 ;TEST START LOCATION
015050 000000      0      ;LOOP ON ERROR START LOCATION
015052 070052      25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
015054 012737 001015 177746      40$:  MOV #OFF,CCR   ;LOOP ON ERROR END LOCATION
015062 004437 002352      JSR R4,RELCTH ;DISABLE CACHE
015066 015222      10$+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

3019 015070 012737 000001 047332      MOV #1,FLT PAT ;1ST FLOATING 1 PATTERN: 000001

```

```

3020 015076 013737 047332 060000      2$:  MOV FLTPAT,60000      ;FLOATING PATTERN TO MAIN MEMORY
3021 015104 112737 000002 177750      1$:  MOVB #HODO,CMR        ;ALLOWS CACHE DATA STORE BITS TO BE
3022                                     ;WRITTEN TO CDR<15:07> ONLY DURING THE
3023                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3024 015112 012737 000015 177746      MOV #15,CCR        ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3025 015120 005737 040000      TST 40000
3026 015124 005737 060000      TST 60000
3027                                     ;WRITE FLOATING 1 PATTERN TO DATA STORE
3028                                     ;LOCATION 0000 FROM MAIN MEMORY
3029 015130 005737 060000      TST 60000        ;LOC. 60000
3030                                     ;WRITE DATA STORE BITS FROM
3031 015134 013737 177754 047316      MOV CDR,CDR150    ;LOC. 0000 INTO CDR<15:0>.
3032                                     ;SAVE CDR CONTENTS
3032 015142 000240      25$:  NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
3032 015144 000240      NOP              ;FOR LOOP ON ERROR
3033 015146 105037 177750      CLRB CMR          ;DISABLE MAINTENANCE
3034 015152 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
3035 015160 023737 047332 047316      CMP FLTPAT,CDR150 ;CHECK FOR CORRECT PATTERN
3036 015166 001410      BEQ 9$           ;PASS
3037 015170 013737 047332 047320      MOV FLTPAT,EXDAT6 ;SAVE FLOATING PATTERN FOR TYPEOUT
3038 015176 104000      ERROR           ;ERROR
3038 015200 015176      .-2            ;-----
3039                                     ; DATA STORE TESTS
3040                                     ;READING CDR<15:0> DID NOT RESULT
3041                                     ;IN CORRECT FLOATING 1 PATTERN
3042 015202 047320      EXDAT6          ;PRINT FLOATING PATTERN EXPECTED
3043 015204 047316      CDR150          ;PRINT CDR<15:0> DATA READ.
3044 015206 000000      0
3045                                     ;:
3046 015210 006337 047332      9$:  ASL FLTPAT     ;NEXT PATTERN
3047 015214 103401      BCS 10$         ;IF PATTERN 100000 HAS BEEN DONE;FINISHED
3048 015216 000727      BR 2$          ;IF NOT, NEXT PASS
3049 015220 000240      10$:  NOP              ;END OF TEST
3050
3051 015222 005237 001464      T110:  INC $TESTN   ;UPDATE TEST ID
3051                                     ;*****
3052                                     ;SBTTL
3053                                     ;SBTTL TEST 110
3054                                     ;SBTTL
3055                                     ;*****
3056                                     ;SBTTL
3057                                     ;SBTTL WRITE ALL 0'S INTO ALL LOW CACHE DATA STORE LOCATIONS (0000 TO 3777).
3058                                     ;SBTTL READ ALL 0'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.
3059                                     ;SBTTL
3059                                     ;*****
3059 015226 000004      SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
3059 015230 015240      40$           ;ERROR/LOOP ON TEST
3059 015232 070024      1$-40$+70000-14 ;TEST START LOCATION
3059 015234 000000      0              ;LOOP ON ERROR START LOCATION
3059 015236 070054      25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
3059 015240 012737 001015 177746      40$:  MOV #OFF,CCR   ;LOOP ON ERROR END LOCATION
3059 015246 004437 002352      JSR R4,RELCTH   ;DISABLE CACHE
3059                                     ;LOCATE TEST CODE TO HIGH CACHE SPACE

```

```

015252 015420          10$+2          :ADDRESS OF START OF NEXT TEST
                                     :THE FOLLOWING LOCATIONS INCLUDING 10$
                                     :ARE RELOCATED TO HI CACHE SPACE

3060 015254 012705 060000          MOV #60000,R5          :ADDRESS 60000 INTO R5
3061 015260 005025          5$: CLR (R5)+             :CLEAR MAIN MEMORY LOW CACHE AREA
3062 015262 020527 070000          CMP R5,#70000         :FINISHED?
3063 015266 001374          BNE 5$                :NO
3064 015270 012705 060000          MOV #60000,R5         :START WITH ADDRESS 60000
3065 015274 012703 040000          MOV #40000,R3         :ADDRESS 40000 INTO R3
3066 015300 112737 000002 177750   1$: MOVB #HODO,CMR       :ALLOWS CACHE DATA STORE BITS TO BE
3067                                     :WRITTEN TO CDR<15:0> ONLY DURING THE
3068                                     :DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3069 015306 012737 000015 177746   MOV #15,CCR           :NO UCB SO AS TO WRITE ENABLE CACHE STORE
3070 015314 005713          TST (R3)              :
3071 015316 005715          TST (R5)              :WRITE ALL 0'S TO DATA STORE FROM MAIN MEM.
3072 015320 005715          TST (R5)              :WRITE DATA STORE BITS INTO CDR<15:0>
3073 015322 013737 177754 047316   MOV CDR,CDR150       :SAVE CDR CONTENTS
3074                                     :
015330 000240          25$: NOP                :INSTRUCTION 'JMP 1$' PLACED HERE
015332 000240          NOP                :FOR LOOP ON ERROR

3075 015334 105037 177750          CLRB CMR              :DISABLE MAINTENANCE
3076 015340 012737 001015 177746   MOV #OFF,CCR          :DISABLE CACHE
3077 015346 022737 000000 047316   CMP #0,CDR150         :CHECK ALL 0'S
3078 015354 001411          BEQ 9$                :PASS
3079 015356 010537 047302          MOV R5,CA210+2       :SAVE ADDRESS USED THIS PASS
3080 015362 005037 047300          CLR CA210
3081                                     :
015366 104000          ERROR                :ERROR
015370 015366          .-2                :-----

3082                                     : DATA STORE TESTS
3083                                     :READING CDR<15:0> DID NOT RESULT
3084                                     :IN ALL 0'S
3085 015372 047300          CA210                :PRINT CA<21:0> USED
3086 015374 047316          CDR150                :PRINT CDR<15:0> DATA READ.
3087 015376 000000          0
3088                                     :
3089 015400 062705 000002          9$: ADD #2,R5           :NEXT CACHE LOCATION
3090 015404 062703 000002          ADD #2,R3
3091 015410 022705 070000          CMP #70000,R5        :HAVE ALL LOW-CACHE LOCATIONS BEEN DONE?
3092 015414 001331          BNE 1$                :NO
3093 015416 000240          10$: NOP                :END OF TEST
3094                                     :
3095 015420 005237 001464          T111: INC $TESTN      :UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 111
.SBTTL
:*****
.SBTTL
.SBTTL WRITE ALL 1'S INTO ALL LOW CACHE DATA STORE LOCATIONS (0000 TO 3777).
.SBTTL READ ALL 1'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.
.SBTTL
:*****
3096
3097
3098
3099
3100
3101

```



```

3102 015424 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          015426 015436          40$          ;ERROR/LOOP ON TEST
          015430 070026          1$-40$+70000-14 ;TEST START LOCATION
          015432 000000          0          ;LOOP ON ERROR START LOCATION
          015434 070056          25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          015436 012737 001015 177746 40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
          015444 004437 002352          JSR R4,RELCTH ;DISABLE CACHE
          015450 015620          10$+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

```

```

3103 015452 012705 060000          MOV #60000,R5          ;ADDRESS 60000 INTO R5
3104 015456 012725 177777          5$: MOV #-1,(R5)+      ;1'S TO MAIN MEMORY LOW CACHE AREA
3105 015462 020527 070000          CMP R5,#70000        ;FINISHED?
3106 015466 001373          BNE 5$              ;NO
3107 015470 012705 060000          MOV #60000,R5        ;START WITH ADDRESS 60000
3108 015474 012703 040000          MOV #40000,R3        ;ADDRESS 40000 INTO R3
3109 015500 112737 000002 177750 1$: MOVB #HODO,CMR      ;ALLOWS CACHE DATA STORE BITS TO BE
3110          ;WRITTEN TO CDR<15:0> ONLY DURING THE
3111          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3112 015506 012737 000015 177746          MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3113 015514 005713          TST (R3)            ;
3114 015516 005715          TST (R5)            ;WRITE ALL 1'S TO DATA STORE FROM MAIN MEM.
3115 015520 005715          TST (R5)            ;WRITE DATA STORE BITS INTO CDR<15:0>
3116 015522 013737 177754 047316          MOV CDR,CDR150      ;SAVE CDR CONTENTS
3117          ;
          015530 000240          25$: NOP            ;INSTRUCTION 'JMP 1$' PLACED HERE
          015532 000240          NOP                ;FOR LOOP ON ERROR
3118 015534 105037 177750          CLRB CMR            ;DISABLE MAINTENANCE
3119 015540 012737 001015 177746          MOV #OFF,CCR        ;DISABLE CACHE
3120 015546 022737 177777 047316          CMP #-1,CDR150      ;CHECK ALL 1'S
3121 015554 001411          BEQ 9$              ;PASS
3122 015556 010537 047302          MOV R5,CA210+2      ;SAVE ADDRESS USED THIS PASS
3123 015562 005037 047300          CLR CA210
3124          015566 104000          ERROR              ;ERROR
          015570 015566          .-2                ;-----
3125          ; DATA STORE TESTS
3126          ;READING CDR<15:0> DID NOT RESULT
3127          ;IN ALL 1'S
3128 015572 047300          CA210              ;PRINT CA<21:0> USED
3129 015574 047316          CDR150             ;PRINT CDR<15:0> DATA READ.
3130 015576 000000          0
3131          ;:::
3132 015600 062705 000002          9$: ADD #2,R5          ;NEXT CACHE LOCATION
3133 015604 062703 000002          ADD #2,R3
3134 015610 022705 070000          CMP #70000,R5       ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
3135 015614 001331          BNE 1$              ;NO
3136 015616 000240          10$: NOP            ;END OF TEST
3137
3138
3139 015620 005237 001464          1112: INC $TESTN    ;UPDATE TEST ID
          ;:*****

```

```

.SBTTL
.SBTTL TEST 112
.SBTTL
:*****
3140 .SBTTL
3141 .SBTTL WRITE ALL 0'S INTO ALL HIGH CACHE DATA STORE LOCATIONS (4000 TO 7777).
3142 .SBTTL READ ALL 0'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.
3143 .SBTTL
3144 .SBTTL
3145 :*****
3146
3147 015624 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$:
015626 015636          40$
015630 060024          1$-40$+60000-14
015632 000000          0
015634 060054          25$-40$+60000-14
015636 012737 001015 177746 40$: MOV #OFF,CCR
015644 004437 002324      JSR R4,RELCTL
015650 016016          10$+2

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

3148 015652 012705 070000          MOV #70000,R5          ;ADDRESS 70000 INTO R5
3149 015656 005025          5$: CLR (R5)+          ;CLEAR MAIN MEMORY HI CACHE AREA
3150 015660 020527 100000          CMP R5,#100000        ;FINISHED?
3151 015664 001374          BNE 5$              ;NO
3152 015666 012705 070000          MOV #70000,R5        ;START WITH ADDRESS 70000
3153 015672 012703 050000          MOV #50000,R3        ;ADDRESS 50000 INTO R3
3154 015676 112737 000002 177750 1$: MOVB #HODO,CMR        ;ALLOWS CACHE DATA STORE BITS TO BE
3155                                     ;WRITTEN TO CDR<15:0> ONLY DURING THE
3156                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3157 015704 012737 000015 177746          MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3158 015712 005713          TST (R3)
3159 015714 005715          TST (R5)
3160 015716 005715          TST (R5)
3161 015720 013737 177754 047316          MOV CDR,CDR150      ;WRITE ALL 0'S TO DATA STORE FROM MAIN MEM.
3162                                     ;WRITE DATA STORE BITS INTO CDR<15:0>
3163                                     ;SAVE CDR CONTENTS
                                25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP          ;FOR LOOP ON ERROR

3163 015732 105037 177750          CLRB CMR            ;DISABLE MAINTENANCE
3164 015736 012737 001015 177746          MOV #OFF,CCR        ;DISABLE CACHE
3165 015744 022737 000000 047316          CMP #0,CDR150       ;CHECK ALL 0'S
3166 015752 001411          BEQ 9$             ;PASS
3167 015754 010537 047302          MOV R5,CA210+2      ;SAVE ADDRESS USED THIS PASS
3168 015760 005037 047300          CLR CA210

                                ERROR          ;ERROR
                                .-2          ;-----

3170                                     ; DATA STORE TESTS
3171                                     ;READING CDR<15:0> DID NOT RESULT
3172 :::
3173                                     ;IN ALL 0'S
3174 015770 047300          CA210              ;PRINT CA<21:0> USED
3175 015772 047316          CDR150            ;PRINT CDR<15:0> DATA READ.

```

L 7

```

3176 015774 000000          0
3177          :::
3178 015776 062705 000002    9$:  ADD #2,R5          ;NEXT CACHE LOCATION
3179 016002 062703 000002    ADD #2,R3
3180 016006 022705 100000    CMP #100000,R5      ;HAVE ALL HIGH CACHE LOCATIONS BEEN DONE?
3181 016012 001331          BNE 1$             ;NO
3182 016014 000240          10$: NOP                ...:END OF TEST
3183
3184
3185 016016 005237 001464    T113: INC $TESTN      ;UPDATE TEST ID
;:*****
.SBTTL
.SBTTL TEST 113
.SBTTL
;:*****
3186          .SBTTL
3187          .SBTTL WRITE ALL 1'S INTO ALL HIGH CACHE DATA STORE LOCATIONS (4000 TO 7777).
3188          .SBTTL READ ALL 1'S EACH CACHE LOCATION FROM CACHE DATA REGISTER.
3189          .SBTTL
3190          .SBTTL
3191          ;:*****
3192 016022 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          016024 016034          40$          ;ERROR/LOOP ON TEST
          016026 060026          1$-40$+60000-14 ;TEST START LOCATION
          016030 000000          0          ;LOOP ON ERROR START LOCATION
          015032 060056          25$-40$+60000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          016034 012737 001015 177746    40$: MOV #OFF,CCR      ;LOOP ON ERROR END LOCATION
          016042 004437 002324          JSR R4,RELCTL      ;DISABLE CACHE
          016046 016716          10$+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

3193 016050 012705 070000    5$:  MOV #70000,R5      ;ADDRESS 70000 INTO R5
3194 016054 012725 177777    MOV #-1,(R5)+      ;1'S TO MAIN MEMORY HI CACHE AREA
3195 016060 020527 100000    CMP R5,#100000    ;FINISHED?
3196 016064 001373          BNE 5$            ;NO
3197 016066 012705 070000    MOV #70000,R5      ;START WITH ADDRESS 70000
3198 016072 012703 050000    MOV #50000,R3      ;ADDRESS 50000 INTO R3
3199 016076 112737 000002 177750    1$:  MOVB #HODO,CMR    ;ALLOWS CACHE DATA STORE BITS TO BE
3200          ;WRITTEN TO CDR<15:0> ONLY DURING THE
3201          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3202 016104 012737 000015 177746    MOV #15,CCR        ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3203 016112 005713          TST (R3)
3204 016114 005715          TST (R5)
3205 016116 005715          TST (R5)
3206 016120 013737 177754 047316    MOV CDR,CDR150    ;WRITE ALL 1'S TO DATA STORE FROM MAIN MEM.
3207          ;WRITE DATA STORE BITS INTO CDR<15:0>
          ;SAVE CDR CONTENTS
          016126 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          016130 000240          NOP          ;FOR LOOP ON ERROR

3208 016132 105037 177750    CLRB CMR          ;DISABLE MAINTENANCE
3209 016136 012737 001015 177746    MOV #OFF,CCR      ;DISABLE CACHE
3210 016144 022737 177777 047316    CMP #-1,CDR150    ;CHECK ALL 1'S
3211 016152 001411          BEQ 9$            ;PASS
3212 016154 010537 047302    MOV R5,CA210+2    ;SAVE ADDRESS USED THIS PASS
3213 016160 005037 047300    CLR CA210
    
```

3214

016164 104000

016166 016164

ERROR

.-2

:ERROR

:-----

```

3216                                     ; DATA STORE TESTS
3217                                     ;READING CDR<15:0> DID NOT RESULT
3218                                     ;IN ALL 1'S
3219 016170 047300                       CA210
3220 016172 047316                       CDR150
3221 016174 00C 00                       0
3222                                     ;:
3223 016176 062705 000002                9$: ADD #2,R5                ;NEXT CACHE LOCATION
3224 016202 062703 000002                ADD #2,R3
3225 016206 022705 100000                CMP #100000,R5           ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
3226 016212 001331                       BNE 1$                  ;NO
3227 016214 000240                       10$: NOP                 ;END OF TEST
3228
3229
3230 016216 005237 001464                T114: INC $TESTN        ;UPDATE TEST ID
3231                                     ;:*****
3232                                     ;.SBTTL
3233                                     ;.SBTTL TEST 114
3234                                     ;.SBTTL
3235                                     ;:*****
3236                                     ;.SBTTL
3237                                     ;.SBTTL VERIFY CACHE DATA STORE ADDRESS LINES (CA<12:1>)
3238                                     ;.SBTTL
3239                                     ;.SBTTL
3240                                     ;.SBTTL
3241 016222 000004                       ;:*****
3242                                     ;SLPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
3243                                     ;40$                    ;ERROR/LOOP ON TEST
3244                                     ;1$-40$+70000-14       ;TEST START LOCATION
3245                                     ;0                      ;LOOP ON ERROR START LOCATION
3246                                     ;25$-40$+70000-14     ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
3247 016224 016234                       40$: MOV #OFF,CCR        ;LOOP ON ERROR END LOCATION
3248 016226 070040                       JSR R4,RELCTH          ;DISABLE CACHE
3249 016230 000000                       10$:+2                 ;LOCATE TEST CODE TO HIGH CACHE SPACE
3250 016232 070100                       ;ADDRESS 0 START OF NEXT TEST
3251 016234 012737 001015 177746
3252 016242 004437 002352
3253 016246 016440
3254                                     ;:THE FOLLOWING LOCATIONS INCLUDING 10$
3255                                     ;:ARE RELOCATED TO HI CACHE SPACE
3256 016250 000240                       NOP
3257                                     ;:
3258                                     ;WHEN THE TEST IS RELOCATED,ADDRESS 70000
3259                                     ;:WILL BE LOADED WITH ALL 1'S DURING TEST
3260 016252 005037 060000                 CLR 60000              ;ALL 0'S TO MAIN MEMORY
3261 016256 012737 00J002 047332         2$: MOV #2,FLTPAT      ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
3262 016264 012702 040000                 MOV #40000,R2
3263 016270 012703 060000                 MOV #60000,R3
3264 016274 063702 047332                 ADD FLTPAT,R2
3265 016300 063703 047332                 ADD FLTPAT,R3
3266 016304 012713 177777                 MOV #-1,(R3)
3267                                     ;:
3268                                     ;R2 CONTAINS 40000+FLTPAT
3269                                     ;R3 CONTAINS 60000+FLTPAT
3270 016310 112737 000002 177750         1$: MOV #HODO,CMR      ;ALL 1'S TO MAIN MEM. LOCATION
3271 016316 012737 000015 177746         MOV #15,CCR           ;SPECIFIED BY R3
3272                                     ;:
3273                                     ;ALLOWS CACHE DATA STORE BITS TO BE
3274                                     ;:WRITTEN TO CDR<15:0> ONLY DURING THE
3275                                     ;:DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3276                                     ;:NO UCB SO AS TO WRITE ENABLE CACHE STORE

```

```

3257 016324 005737 040000          TST 40000
3258 016330 005737 060000          TST 60000
3259                                     ;WRITE ALL 0'S FROM MAIN MEM. LOCATION
3260 016334 005712                                     ;60000 INTO CACHE DATA STORE LOCATION 0000.
3261 016336 005713          TST (R2)
3262                                     TST (R3)
3263                                     ;WRITE ALL 1'S FROM MAIN MEM. INTO
3264 016340 005737 060000          TST 60000          ;CACHE DATA STORE LOCAT. SPECIFIED
3265                                     ;BY R3'S BITS 1 THRU 12: CA<12:1>
3266 016344 013701 177754          MOV CDR,R1          ;LOAD DATA FROM CACHE DATA STORE LOCATION
3267                                     ;0000 INTO CDR.
          016350 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          016352 000240          NOP          ;FOR LOOP ON ERROR

3268 016354 105037 177750          CLRB CMR          ;DISABLE MAINT. MODE
3269 016360 012737 001015 177746          MOV #OFF,CCR      ;DISABLE CACHE
3270 016366 005701          TST R1          ;CHECK FOR ALL 0'S
3271 016370 001411          BEQ 9$          ;PASS
3272
3273 016372 013737 047332 047322          MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
3274 016400 006237 047322          ASR CA121          ;PREPARE CA121 FOR TYPEOUT
3275
          016404 104000          ERROR          ;ERROR
          016406 016404          .-2          ;-----

3276                                     ;DATA STORE TESTS- ADDRESS LINE VERIFICATION
3277                                     ;READING CDR<15:0> FOR CACHE DATA STORE
3278                                     ;DID NOT RESULT IN ALL 0'S
3279 016410 047322          CA121          ;PRINT CACHE DATA STORE ADDRESS LOCATION
3280                                     ;USED: CA<12:1>. NOTE THAT THE 1 IN
3281                                     ;THIS PATTERN WILL POINT TO THE ADDRESS
3282                                     ;LINE THAT POSSIBLY CAUSES ERROR.
3283 016412 000000          0
3284
3285 016414 006337 047332          9$: ASL FLTPAT          ;...
3286 016420 022737 020000 047332          CMP #20000,FLTPAT ;NEXT PATTERN
3287 016426 001316          BNE 2$          ;HAS CACHE DATA STORE LOCAT. 4000 BEEN DONE?
3288 016430 012737 000240 070000          MOV #240,70000    ;NO
3289                                     ;RESTORE OVERWRITTEN LOCATION 7000
3290 016436 000240          10$: NOP          ;WITH A NOP
3291                                     ;END OF TEST
3292
3293
3294
3295
3296 .SBTTL *
3297 .SBTTL *
3298 .SBTTL *          COMPARE TAG STORE DATA WITH CACHE ADDRESS LINE DATA TESTS
3299 .SBTTL *
3300 .SBTTL *
3301
3302
3303
3304
3305
3306 016440 005237 001464          T115: INC $TESTN          ;UPDATE TEST ID
          ;:*****

```

```

.SBTTL
.SBTTL TEST 115
.SBTTL
:;*****
3307 .SBTTL
3308 .SBTTL
3309 .SBTTL VERIFY THAT AN EQUAL DATA COMPARISON CONDITION CAN EXIST
3310 .SBTTL BY COMPARING TAG STORE DATA AND CA<21:13>
3311 .SBTTL
3312 .SBTTL UNDER THE FOLLOWING CONDITIONS CMR<15:13> SHOULD RESULT IN ALL
3313 .SBTTL 1'S INDICATING A MATCH :
3314 .SBTTL
3315 .SBTTL TAG STORE DATA AND CA<21:13> ALL 0'S
3316 .SBTTL
3317 .SBTTL
:;*****
3318 016444 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
016446 016456 40$ ;TEST START LOCATION
016450 070000 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
016452 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
016454 070036 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
016456 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
016464 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
016470 016634 10$+2 ;ADDRESS OF START OF NEXT TEST

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

3319 016472 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS COMPARED RESULTS TO BE
3320 ;WRITTEN TO CMR<15:13> ONLY DURING
3321 ;THE DESTINATION MEMORY ACCESS
3322 ;OF AN INSTRUCTION
3323 016500 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3324 016506 005737 040000 TST 40000
3325 016512 005737 000000 TST 0 ;WRITE ALL 0'S INTO TAG STORE LOCATION 0000
3326 ;FROM CACHE ADDRESS CA<21:13>
3327 016516 005737 000000 TST 0 ;PLACE ALL 0'S ON CA<21:13> FOR COMPARISON
3328 ;WITH TAG STORE DATA. WRITE COMPARED
3329 ;RESULTS INDICATION IN CMR<15:13>
3330 016522 013737 177750 047326 MOV CMR,CM1513 ;SAVE CMR DATA
3331 ;INSTRUCTION 'JMP 1$' PLACED HERE
016530 000240 25$: NOP ;FOR LOOP ON ERROR
016532 000240 NOP

3332 016534 105037 177750 CLRB CMR ;DISABLE MAINTENANCE MODE
3333 016540 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3334 016546 042737 017777 047326 BIC #17777,CM1513
3335 016554 022737 160000 047326 CMP #160000,CM1513 ;CHECK THAT CMR<15:13> ALL 1'S
3336 016562 001423 BEQ 10$ ;PASS
3337
3338
3339 016564 012737 000007 047324 MOV #7,EXDAT1 ;INDICATE EXPECTED CMR<15:13>
3340 016572 012737 000015 002052 MOV #13,LOOP ;ERROR;PREPARE CM1513. FOR TYPEOUT
3341 016600 006237 047326 2$: ASR CM1513
3342 016604 042737 100000 047326 BIC #100000,CM1513
3343 016612 005337 002052 DEC LOOP
3344 016616 001370 BNE 2$
3345

```

```

016620 104000          ERROR          ;ERROR
                                ;-----
016622 016620          .-2
3346                                     ;COMPARE TAG STORE & CA<21:13> TESTS
3347                                     ;BITS 15 THRU 13 OF CMR DID NOT READ
3348                                     ;AS ALL 1'S
3349 016624 047324     EXDAT1         ;PRINT CMR<15:13> DATA EXPECTED
3350 016626 047326     CM1513        ;PRINT CMR<15:13> DATA RECEIVED
3351 016630 000000     0
3352 016632 000240     10$: NOP      ;END OF TEST
3353
3354
3355
3356

```

```

3357 016634 005237 001464 T116:          INC $TESTN          ;UPDATE TEST ID
;*****

```

```

3358 .SBTTL
3359 .SBTTL TEST 116
3360 .SBTTL
;*****
3361 .SBTTL
3362 .SBTTL VERIFY THAT AN UNEQUAL DATA COMPARISON CONDITION CAN BE DETECTED
3363 .SBTTL BY COMPARING TAG STORE AND CA<21:13>
3364 .SBTTL
3365 .SBTTL UNDER THE FOLLOWING CONDITIONS FOR TAG STORE DATA AND CA<21:13>
3366 .SBTTL CMR<15:13> SHOULD READ AS SPECIFIED:

```

```

3367 JMP 38$+2
3368
3369 016640 000137 016666
;::: CMR<15:13>          CA<21:13>          TAG STORE
;::: -----
3369 30$:          6          : 001          000
3370 31$:          5          : 002          000
3371 32$:          5          : 004          000
3372 33$:          5          : 010          000
3373 34$:          5          : 020          000
3374 35$:          3          : 040          000
3375 36$:          3          : 100          000
3376 37$:          3          : 200          000
3377 38$:          3          : 400          000

```

```

3378 .SBTTL
3379 .SBTTL
;*****
3381 016666 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
016670 016700          40$          ;TEST START LOCATION
016672 070012          1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
016674 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
016676 070104          25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
016700 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
016706 004437 002352          JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
016712 017230          10$+2          ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

3382 016714 012737 000200 172350 2$: MOV #200,KPAR4 ;KPAR4 CONTAINS THE FLOATING 1 PATTERN
3383                                     ;AND REPRESENTS THE THE PAGE ADDRESS FIELD

```



```

3384
3385
3386
3387
3388 016722 012701 016644          MOV #30$,R1
3389
3390 016726 012737 070254 000004    1$:  MOV #7$-2$+70000,4
3391 016734 012737 000340 000006    MOV #340,6
3392 016742 112737 000002 177750    MOV# #HODO,CMR
3393
3394
3395 016750 012737 000015 177746    MOV #15,CCR
3396 016756 012737 000001 177572    MOV #1,SRO
3397 016764 012737 000020 172516    MOV #20,SR3
3398 016772 005737 040000            *ST 40000
3399 016776 005737 000000            TST 0
3400 017002 005737 100000            TST 100000
3401
3402
3403
3404
3405
3406
3407 017006 000240                    NOP
3408 017010 000240                    NOP
3409 017012 013737 177750 047326    MOV CMR,CM1513
3410
      017020 000240                    25$:  NOP
      017022 000240                    NOP
3411 017024 005037 177572            CLR SRO
3412 017030 005037 172516            CLR SR3
3413 017034 012737 001015 177746    MOV #OFF,CCR
3414 017042 105037 177750            CLRB CMR
3415 017046 012737 000015 002052    MOV #13,LOOP
3416 017054 006237 047326            5$:  ASR CM1513
3417 017060 042737 100000 047326    BIC #100000,CM1513
3418 017066 005337 002052            DEC LOOP
3419 017072 001370                    BNE 5$
3420 017074 023711 047326            CMP CM1513,(R1)
3421
3422 017100 001426                    BEQ 9$
3423 017102 013737 172350 047312    MOV KPAR4,CA2113
3424 017110 012737 000007 002052    MOV #7,LOOP
3425 017116 006237 047312            6$:  ASR CA2113
3426 017122 042737 100000 047312    BIC #100000,CA2113
3427 017130 005337 002052            DEC LOOP
3428 017134 001370                    BNE 6$
3429 017136 011137 047324            MOV (R1),EXDAT1
3430
      017142 104000                    ERROR
      017144 017142                    .-2
3431
3432
3433 017146 047312                    CA2113
3434

```

```

;DATA USED BY MEMORY MNGMNT. TO CONSTRUCT
;THE PHYSICAL ADDRESS. 200 IS THE 1ST
;FLOATING 1 PATTERN WHICH WILL BE CONSTRUCTED
;TO ADDRESS 20000.
;SAVE ADDRESS OF FIRST CMR<15:13>
;EXPECTED DATA
;ALLOW FOR NEX TRAP
;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 CACHE STORE
;ENABLE MEMORY MNGMNT.
;ENABLE 22-BIT MAPPING
;
;WRITE ALL 0'S TO TAG STORE LOCATION 0000
;CHOOSES KPAR4 FOR ADDRESSING. PHYSICAL
;ADDRESS WILL BE DETERMINED BY FLOATING
;PATTERN USED IN KPAR4.
;PLACE DLOATING PATTERN ON CA<21:13>
;FOR COMPARISON WITH TAG STORE DATA
;AT LOCATION 0000. WRITE THE COMPARED
;RESULT INDICATION INTO CMR<15:13>.
;SAVE CMR DATA
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;DISABLE MEM MNGMENT.
;DISABLE CACHE
;DISABLE MAINTENANCE
;PREPARE CM1513. FOR ERROR CHECK
;COMPARE CMR<15:13> RECEIVED WITH
;EXPECTED
;PASS
;SAVE PATTERN USED FOR CA<21:13>
;PREPARE CA2113 FOR PRINTOUT
;PREPARE EXPECTED FOR PRINTOUT
;ERROR
;-----
;COMPARE TAG STORE AND CA<21:13> TESTS
;CMR<15:13> DID NOT READ CORRECTLY
;PRINT CA<21:13> PATTERN USED ON ADDRESS
;LINES

```

```

3435 017150 047324          EXDAT1          ;PRINT CMR<15:13> EXPECTED
3436 017152 047326          CM1513          ;PRINT CMR<15:13> RECEIVED
3437 017154 000000          0              ;PRINT TERMINATE
3438 017156 006337 172350   9$:  ASI KPAR4   ;NEXT PATTERN;IF PHYSICAL ADDRESS
3439                                ;1000000 HAS BEEN DONE; FINISHED
3440 017162 103414          BCS 8$         ;
3441 017164 005721          TST (R1)+      ;NOT FINISHED; POINT TO NEXT
3442                                ;CMR<15:13> EXPECTED
3443 017166 000657          BR 1$          ;CONTINUE WITH TEST
3444 017170 005037 177572   7$:  CLR SR0     ;DISABLE MEM. MNGMNT.
3445 017174 005037 172516   CLR SR3
3446 017200 105037 177750   CLR#B CMR      ;DISABLE MAINTENANCE
3447 017204 012737 001015 177746  MOV #OFF,CCR   ;DISABLE CACHE
3448 017212 022626          CMP (SP)+,(SP)+;RESTORE STACK DUE TO INTERRUPT
3449 017214 012737 000006 000004 8$:  MOV #6,4    ;RESTORE VECTORS
3450 017222 005037 000006          CLR 6
3451 017226 000240          10$: NOP        ;END OF TEST
3452
3453
3454
3455
3456          .SBTTL *
3457          .SBTTL
3458          .SBTTL *          FLUSH CACHE TESTS
3459          .SBTTL
3460          .SBTTL *
3461
3462
3463
3464
3465 017230 005237 001464   T117:          INC $TESTN          ;UPDATE TEST ID
;:*****
;:          .SBTTL
;:          .SBTTL TEST 117
;:          .SBTTL
;:          .SBTTL
;:          .SBTTL VERIFY FLUSH IN PROGRESS BIT WILL SET AS A RESULT OF FLUSH
;:          .SBTTL
;:*****
3466
3467
3468
3469
3470 017234 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;:          ;ERROR/LOOP ON TEST
;:          ;TEST START LOCATION
;:          ;LOOP ON ERROR START LOCATION
;:          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;:          ;LOOP ON ERROR END LOCATION
;:          40$:
;:          1$: CLR R2          ;INITIALIZE COUNTER
;:          CLR FAIL1      ;INITIALIZE ERROR FLAG
;:          BIS #FC,CCR    ;FLUSH CACHE
;:          BIT #VCIP,CCR  ;VERIFY FLUSH IN PROGRESS
;:          BNE 3$         ;VCIP BIT IS SET
;:          INC FAIL1     ;INDICATE ERROR
;:          DEC R2        ;WAIT DELAY FOR FLUSH TO COMPLETE
;:          BNE .-2
;:          3$:
;:          25$: NOP        ;INSTRUCTION 'JMP 1$' PLACED HERE
;:          NOP          ;FOR LOOP ON ERROR

```

```

3480 017306 005737 002060      TST FAIL1      ;IS THERE ERROR
3481 017312 001403              BEQ 10$        ;PASS
3482      017314 104000          ERROR          ;ERROR
                                   ;-----
      017316 017314          .-2
3483                               ;FLUSH CACHE TESTS
3484                               ;FLUSH IN PROGRESS BIT(VCIP) FAILED
3485                               ;TO SET AS A RESULT OF SETTING CACHE FLUSH BIT
3486 017320 000000              0
3487 017322 000240          10$: NOP          ;END OF TEST
3488
3489
3490
3491

```

```

3492 017324 005237 001464      T120:          INC $TESTN      ;UPDATE TEST ID

```

```

:*****

```

```

3493      .SBTTL
3494      .SBTTL TEST 120
3495      .SBTTL
3496      .SBTTL
3497      .SBTTL VERIFY FLUSH IN PROGRESS BIT(VCIP) WILL RESET ON COMPLETION OF FLUSH
3498      .SBTTL

```

```

3499 017330 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      017332 017342          40$              ;ERROR/LOOP ON TEST
      017334 017342          1$              ;TEST START LOCATION
      017336 000000          0                ;LOOP ON ERROR START LOCATION
      017340 017366          25$             ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      017342          40$:
3498 017342 005002          1$: CLR R2          ;INITIALIZE COUNTER
3499 017344 052737 000400 177746          BIS #FC,CCR    ;START FLUSH
3500 017352 032737 010000 177746          3$: BIT #VCIP,CCR ;SEE IF FLUSH COMPLETE
3501 017360 001407          BEQ 10$        ;FLUSH COMPLETE
3502 017362 005302          DEC R2        ;SEE IF TIME HAS RUN OUT
3503 017364 001372          BNE 3$        ;NOT YET
3504      017366 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
      017370 000240          NOP          ;FOR LOOP ON ERROR

```

```

3505 017372 104000          ERROR          ;ERROR
                                   ;-----
      017374 017372          .-2
3506                               ;FLUSH CACHE TESTS
3507                               ;FLUS IN PROGRESS BIT FAILED TO CLEAR
3508                               ;TIME FOR FLUSH TO COMPLETE RAN OUT
3509 017376 000000              0
3510 017400 000240          10$: NOP          ;END OF TEST
3511
3512
3513
3514

```

```

3515 017402 005237 001464      T121:          INC $TESTN      ;UPDATE TEST ID

```

```

:*****

```

```

3516
3517
3518
3519
3520
3521 017406 000004
      017410 017420
      017412 017420
      017414 000000
      017416 017464
      017420
3522 017420 032737 020000 177746
3523 017426 001407
3524 017430 052737 000400 177746
3525 017436 032737 010000 177746
3526 017444 001374
3527 017446 052737 000400 177746
3528 017454 032737 010000 177746
3529 017462 001374
3530
      017464 000240
      017466 000240
3531 017470 032737 020000 177746
3532
3533 017476 001003
3534
      017500 104000
      017502 017500
3535
3536
3537 017504 000000
3538 017506 000240
3539
3540
3541
3542
3543 017510 005237 001464

```

```

.SBTTL
.SBTTL TEST 121
.SBTTL
.SBTTL
.SBTTL VERIFY THAT VSIU BIT WILL CHANGE FROM A CLEAR TO SET CONDITION AS
.SBTTL A RESULT OF CACHE FLUSH
.SBTTL
:*****
      SCPCND
      40$
      1$
      0
      25$
      40$:
      1$: BIT #VSIU,CCR
      BEQ 3$
      BIS #FC,CCR
      BIT #VCIP,CCR
      BNE -6
      3$: BIS #FC,CCR
      BIT #VCIP,CCR
      BNE -6
      25$: NOP
      NOP
      BIT #VSIU,CCR
      BNE 10$
      ERROR
      -2
      10$: 0
      NOP
      T122: 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
:IS SET A BEING USED
:YES
:CAUSE FLUSH FOR SET A
:WAIT FOR FLUSH TO COMPLETE
:CAUSE FLUSH
:WAIT FOR FLUSH TO COMPLETE
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:IS VSIU BIT =1 INDICATING VALID SET
:B WAS SELECTED
:PASS
:ERROR
:-----
:FLUSH CACHE TESTS
:VSIU BIT DID NOT SET AS A RESULT OF FLUSH
:END OF TEST
:UPDATE TEST ID

```

```

3544
3545
3546
3547
3548
3549 017514 000004
      017516 017526
      017520 017526
      017522 000000

```

```

.SBTTL
.SBTTL TEST 122
.SBTTL
.SBTTL
.SBTTL VERIFY THAT VSIU BIT WILL CHANGE FROM A SET TO CLEAR CONDITION AS
.SBTTL A RESULT OF CACHE FLUSH
.SBTTL
:*****
      SCPCND
      40$
      1$
      0

```

```

: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

```

```

017524 017572          25$          ;LOOP ON ERROR END LOCATION
017526          40$:          ;
3550 017526 032737 020000 177746 1$:          BIT #VSIU,CCR          ;IS SET B BEING USED
3551 017534 001007          BNE 3$          ;YES
3552 017536 052737 000400 177746  BIS #FC,CCR          ;CAUSE FLUSH FOR SET B
3553 017544 032737 010000 177746  BIT #VCIP,CCR        ;WAIT FOR FLUSH TO COMPLETE
3554 017552 001374          BNE .-6
3555 017554 052737 000400 177746  3$:          BIS #FC,CCR          ;CAUSE FLUSH
3556 017562 032737 010000 177746  BIT #VCIP,CCR        ;WAIT FOR FLUSH TO COMPLETE
3557 017570 001374          BNE .-6
3558          .25$:          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          017572 000240          NOP          ;FOR LOOP ON ERROR
          017574 000240
3559 017576 032737 020000 177746  BIT #VSIU,CCR        ;IS VSIU BIT =0 INDICATING VALID SET
3560          BEQ 10$          ;A WAS SELECTED
3561 017604 001403          ;PASS
3562          017606 104000          ;ERROR
          017610 017606          ;-----
          .-2          ;FLUSH CACHE TESTS
3563          ;VSIU BIT DID NOT CLEAR AS A RESULT OF FLUSH
3564          0
3565 017612 000000          10$:          NOP          ;END OF TEST
3566 017614 000240
3567
3568
3569
3570          .SBTTL *
3571          .SBTTL
3572          .SBTTL *
3573          .SBTTL          VALID BITS STORE TESTS- SET A
3574          .SBTTL *
3575
3576
3577
3578 017616 005237 001464  T123:          INC $TESTN          ;UPDATE TEST ID
          ;*****
          .SBTTL
          .SBTTL TEST 123
          .SBTTL
          .SBTTL WRITE AND READ 0'S TO ALL LOW CACHE VALID
          .SBTTL BIT STORE ADDRESS LOCATIONS- SET A
          .SBTTL (VALID STORE LOCATIONS 0000 TO 3777)
          .SBTTL
          ;*****
3579          .SBTTL
3580          .SBTTL
3581          .SBTTL
3582          .SBTTL
3583          .SBTTL
3584 017622 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          017624 017634          40$          ;TEST START LOCATION
          017626 070052          1$-40$+70000-14      ;LOOP ON ERROR START LOCATION
          017630 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          017632 070100          25$-40$+70000-14      ;LOOP ON ERROR END LOCATION
          017634 012737 001015 177746  40$:          MOV #OFF,CCR          ;DISABLE CACHE
          017642 004437 002352          JSR R4,RELCTH        ;LOCATE TEST CODE TO HIGH CACHE SPACE
          017646 020034          10$+2          ;ADDRESS OF START OF NEXT TEST

```

;THE FOLLOWING LOCATIONS INCLUDING 10\$

:ARE RELOCATED TO HI CACHE SPACE

3585	017650	032737	020000	177746		BIT #VSIU,CCR	:IS SET A BEING USED?
3586	017656	001407				BEQ 3\$:YES
3587	017660	052737	000400	177746		BIS #FC,CCR	:NO; FLUSH CACHE FOR SET A
3588	017666	032737	010000	177746		BIT #VCIP,CCR	:WAIT TILL FLUSH COMPLETE
3589	017674	001374				BNE .-6	
3590	017676	012737	177777	177752	3\$:	MOV #-1,CHR	:LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
3591							:REGISTERS, SINCE TDAR WILL BE USED
3592	017704	112737	000374	177751		MOVB #374,CMR+1	
3593	017712	012705	060000			MOV #60000,R5	::ADDRESS 60000 INTO R5
3594	017716	012703	040000			MOV #40000,R3	:ADDRESS 40000 INTO R3
3595	017722	012737	000015	177746	1\$:	MOV #15,CCR	:NO UCB SO AS TO WRITE ENABLE VALID STORE
3596	017730	112737	000003	177750		MOVB #HODO+TDAR,CMR	:HODO ALLOWS VALID STORE SET A TO
3597							:BE WRITTEN TO CMR<12> ONLY DURING
3598							:THE DESTINATION MEMORY ACCESS.
3599							:TDAR WILL FORCE A 0 TO BE WRITTEN
3600							:INTO VALID STORE WHEN A WRITE TO
3601							:VALID STORE OCCURS
3602	017736	005713				TST (R3)	
3603	017740	005715				TST (R5)	:WRITE A 0 INTO VALID STORE ADDRESS
3604							:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3605	017742	005715				TST (R5)	:WRITE VALID STORE DATA INTO CMR<12>
3606							:FROM VALID STORE ADDRESS LOCATION
3607							:JUST WRITTEN INTO.
3608	017744	013701	177750			MOV CMR,R1	:SAVE CMR DATA
3609							
	017750	000240			25\$:	NOP	:INSTRUCTION 'JMP 1\$' PLACED HERE
	017752	000240				NOP	:FOR LOOP ON ERROR
3610	017754	105037	177750			CLRB CMR	:DISABLE MAINT. MODE
3611	017760	012737	001015	177746		MOV #OFF,CCR	:DISABLE CACHE
3612	017766	032701	010000			BIT #VLD,R1	:CMR<12> SHOULD BE 0.
3613	017772	001410				BEQ 9\$:PASS
3614							
3615	017774	010537	047322			MOV R5,CA121	:SAVE VALID STORE ADDRESS LOCATION
3616							:USED: CA<12:1>
3617	020000	006237	047322			ASR CA121	:PREPARE CA121 FOR TYPEOUT
3618							
	020004	104000				ERROR	:ERROR
							:-----
	020006	020004				.-2	
3619							:VALID BITS STORE TESTS
3620							:READING VALID STORE DATA SET A
3621							:THRU CMR<12> DID NOT RESULT IN 0.
3622	020010	047322				CA121	:PRINT VALID STORE ADDRESS LOCATION
3623							:USED: CA<12:1>.
3624	020012	000000				0	
3625					:::		
3626	020014	062705	000002		9\$:	ADD #2,R5	:NEXT VALID STORE LOCATION
3627	020020	062703	000002			ADD #2,R3	
3628	020024	020527	070000			CMP R5,#70000	:HAVE ALL LOW CACHE ADDRESS LOCATIONS
3629							:BEEN DONE?
3630	020030	001334				BNE 1\$:NO
3631	020032	000240			10\$:	NOP	:END OF TEST
3632							
3633							

```

3634 020034 005237 001464      T124:          INC $TESTN          ;UPDATE TEST ID
;:*****
.SBTTL
.SBTTL TEST 124
.SBTTL
3635 .SBTTL WRITE AND READ 1'S TO ALL LOW CACHE VALID
3636 .SBTTL BIT STORE ADDRESS LOCATIONS- SET A
3637 .SBTTL (VALID STORE LOCATIONS 0000 TO 3777)
3638 .SBTTL
3639 ;:*****
3640 020040 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
020042 020052          40$          ;TEST START LOCATION
020044 070036          1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
020046 000000          0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
020050 070064          25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
020052 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
020060 004437 002352      JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
020064 020236          10$+2        ;ADDRESS OF START OF NEXT TEST

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

3641 020066 032737 020000 177746      BIT #VSIU,CCR ;IS SET A BEING USED?
3642 020074 001407          BEQ 3$        ;YES
3643 020076 052737 000400 177746      BIS #FC,CCR  ;NO; FLUSH CACHE FOR SET A
3644 020104 032737 010000 177746      BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3645 020112 001374          BNE .-6
3646 020114 012705 060000      3$: MOV #60000,R5 ;:ADDRESS 60000 INTO R5
3647 020120 012703 040000      MOV #40000,R3 ;:ADDRESS 40000 INTO R3
3648 020124 012737 000015 177746      1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3649 020132 112737 000002 177750      MOVB #HODO,CMR ;HODO ALLOWS VALID STORE SET A TO
3650 ;BE WRITTEN TO CMR<12> ONLY DURING
3651 ;THE DESTINATION MEMORY ACCESS.
3652 020140 005713          TST (R3)
3653 020142 005715          TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS
3654 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3655 020144 005715          TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
3656 ;FROM VALID STORE ADDRESS LOCATION
3657 ;JUST WRITTEN INTO.
3658 020146 013701 177750      MOV CMR,R1 ;SAVE CMR DATA
3659 ;INSTRUCTION 'JMP 1$' PLACED HERE
020152 000240      25$: NOP ;FOR LOOP ON ERROR
020154 000240      NOP

3660 020156 105037 177750      CLRB CMR ;DISABLE MAINT. MODE
3661 020162 012737 001015 177746      MOV #OFF,CCR ;DISABLE CACHE
3662 020170 032701 010000      BIT #VLD,R1 ;CMR<12> SHOULD BE 1.
3663 020174 001010          BNE 9$        ;PASS
3664
3665 020176 010537 047322      MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
3666 ;USED: CA<12:1>
3667 020202 006237 047322      ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3668
020206 104000          ERROR ;ERROR
020210 020206          .-2 ;-----

```

```

3669                                     ;VALID BITS STORE TESTS
3670                                     ;READING VALID STORE DATA SET A
3671                                     ;THRU CMR<12> DID NOT RESULT IN 1.
3672 020212 047322                       CA121                               ;PRINT VALID STORE ADDRESS LOCATION
3673                                     ;USED: CA<12:1>.
3674 020214 000000                       0
3675                                     ;:
3676 020216 062705 000002                 9$: ADD #2,R5                               ;NEXT VALID STORE LOCATION
3677 020222 062703 000002                 ADD #2,R3
3678 020226 020527 070000                 CMP R5,#70000                               ;HAVE ALL LOW CACHE ADDRESS LOCATIONS
3679                                     ;BEEN DONE?
3680 020232 001334                       BNE 1$                                       ;NO
3681 020234 000240                       10$: NOP                                     ;END OF TEST
3682
3683
3684
3685 020236 005237 001464                 T125: INC $TESTN                               ;UPDATE TEST ID
;:*****
.SBTTL
.SBTTL TEST 125
.SBTTL
.SBTTL WRITE AND READ 0'S TO ALL HIGH CACHE VALID
.SBTTL BIT STORE ADDRESS LOCATIONS- SET A
.SBTTL (VALID STORE LOCATIONS 4000 TO 7777)
.SBTTL
;:*****
3686                                     ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
3687                                     ;ERROR/LOOP ON TEST
3688                                     ;TEST START LOCATION
3689                                     ;LOOP ON ERROR START LOCATION
3690                                     ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
3691 020242 000004                       SCPCND                                       ;LOOP ON ERROR END LOCATION
020244 020254                       40$                                         ;DISABLE CACHE
020246 060052                       1$-40$+60000-14                           ;LOCATE TEST CODE TO LOW CACHE SPACE
020250 000000                       0                                           ;ADDRESS OF START OF NEXT TEST
020252 060100                       25$-40$+60000-14
020254 012737 001015 177746             40$: MOV #OFF,CCR
020262 004437 002324                 JSR R4,RELCTL
020266 020454                       10$+2

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

3692 020270 032737 020000 177746         BIT #VSIU,CCR                               ;IS SET A BEING USED?
3693 020276 001407                       BEQ 3$                                       ;YES
3694 020300 052737 000400 177746         BIS #FC,CCR                                 ;NO; FLUSH CACHE FOR SET A
3695 020306 032737 010000 177746         BIT #VCIP,CCR                               ;WAIT TILL FLUSH COMPLETE
3696 020314 001374                       BNE .-6
3697 020316 012737 177777 177752         3$: MOV #-1,CHR                             ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
3698                                     ;REGISTERS, SINCE TDAR WILL BE USED
3699 020324 112737 000374 177751         MOVB #374,CMR+1
3700 020332 012705 070000                 MOV #70000,R5                               ;:ADDRESS 70000 INTO R5
3701 020336 012703 050000                 MOV #50000,R3                               ;:ADDRESS 50000 INTO R3
3702 020342 012737 000015 177746         1$: MOV #15,CCR                             ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3703 020350 112737 000003 177750         MOVB #HODO+TDAR,CMR                       ;HODO ALLOWS VALID STORE SET A TO
3704                                     ;BE WRITTEN TO CMR<12> ONLY DURING
3705                                     ;THE DESTINATION MEMORY ACCESS.
3706                                     ;TDAR WILL FORCE A 0 TO BE WRITTEN
3707                                     ;INTO VALID STORE WHEN A WRITE TO
3708                                     ;VALID STORE OCCURS
3709 020356 005713                       TST (R3)

```



```

3710 020360 005715          TST (R5)          ;WRITE A 0 INTO VALID STORE ADDRESS
3711                          ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3712 020362 005715          TST (R5)          ;WRITE VALID STORE DATA INTO CMR<12>
3713                          ;FROM VALID STORE ADDRESS LOCATION
3714                          ;JUST WRITTEN INTO.
3715 020364 013701 177750    MOV CMR,R1        ;SAVE CMR DATA
3716                          ;
      020370 000240          25$: NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
      020372 000240          NOP                ;FOR LOOP ON ERROR
3717 020374 105037 177750    CLRB CMR          ;DISABLE MAINT. MODE
3718 020400 012737 001015    MOV #OFF,CCR     ;DISABLE CACHE
3719 020406 032701 010000    BIT #VLD,R1    ;CMR<12> SHOULD BE 0.
3720 020412 001410          BEQ 9$            ;PASS
3721                          ;
3722 020414 010537 047322    MOV R5,CA121    ;SAVE VALID STORE ADDRESS LOCATION
3723                          ;USED: CA<12:1>
3724 020420 006237 047322    ASR CA121      ;PREPARE CA121 FOR TYPEOUT
3725                          ;
      020424 104000          ERROR                ;ERROR
      ;-----
      020426 020424          .-2
3726                          ;VALID BITS STORE TESTS
3727                          ;READING VALID STORE DATA SET A
3728                          ;THRU CMR<12> DID NOT RESULT IN 0.
3729 020430 047322          CA121                ;PRINT VALID STORE ADDRESS LOCATION
3730                          ;USED: CA<12:1>.
3731 020432 000000          0
3732                          ;
3733 020434 062705 000002    ::: 9$: ADD #2,R5    ;NEXT VALID STORE LOCATION
3734 020440 062703 000002    ADD #?,R3
3735 020444 020527 100000    CMP R5,#100000 ;HAVE ALL HIGH CACHE ADDRESS LOCATIONS
3736                          ;BEEN DONE?
3737 020450 001334          BNE 1$            ;NO
3738 020452 000240          10$: NOP                ;END OF TEST
3739                          ;
3740                          ;
3741 020454 005237 001464    T126: INC $TESTN ;UPDATE TEST ID
      ;*****
      .SBTTL
      .SBTTL TEST 126
      .SBTTL
      .SBTTL WRITE AND READ 1'S TO ALL HIGH CACHE VALID
      .SBTTL BIT STORE ADDRESS LOCATIONS- SET A
      .SBTTL (VALID STORE LOCATIONS 4000 TO 7777)
      .SBTTL
      ;*****
3742 020460 000004          SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
3743                          ;ERROR/LOOP ON TEST
3744                          ;TEST START LOCATION
3745                          ;LOOP ON ERROR START LOCATION
3746                          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      020462 020472          40$
      020464 060036          1$-40$+60000-14
      020466 000000          0
      020470 060064          25$-40$+60000-14
      020472 012737 001015    40$: MOV #OFF,CCR    ;DISABLE CACHE
      020500 004437 002324    JSR R4,RELCTL  ;LOCATE TEST CODE TO LOW CACHE SPACE
      020504 020656          10$+2                ;ADDRESS OF START OF NEXT TEST

```

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

```

3748 020506 032737 020000 177746      BIT #VSIU,CCR      ;IS SET A BEING USED?
3749 020514 001407                    BEQ 3$             ;YES
3750 020516 052737 000400 177746      BIS #FC,CCR        ;NO; FLUSH CACHE FOR SET A
3751 020524 032737 010000 177746      BIT #VCIP,CCR      ;WAIT TILL FLUSH COMPLETE
3752 020532 001374                    BNE .-6
3753 020534 012705 070000      3$:  MOV #70000,R5    ;; ADDRESS 70000 INTO R5
3754 020540 012703 050000      MOV #50000,R3     ;ADDRESS 50000 INTO R3
3755 020544 012737 000015 177746      1$:  MOV #15,CCR      ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3756 020552 112737 000002 177750      MOVB #HODO,CMR    ;HODO ALLOWS VALID STORE SET A TO
3757                                ;BE WRITTEN TO CMR<12> ONLY DURING
3758                                ;THE DESTINATION MEMORY ACCESS.
3759 020560 005713                    TST (R3)
3760 020562 005715                    TST (R5)          ;WRITE A 1 INTO VALID STORE ADDRESS
3761                                ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3762 020564 005715                    TST (R5)          ;WRITE VALID STORE DATA INTO CMR<12>
3763                                ;FROM VALID STORE ADDRESS LOCATION
3764                                ;JUST WRITTEN INTO.
3765 020566 013701 177750      MOV CMR,R1        ;SAVE CMR DATA
3766                                ;
020572 000240      25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERF
020574 000240      NOP          ;FOR LOOP ON ERROR
3767 020576 105037 177750      CLRB CMR          ;DISABLE MAINT. MODE
3768 020602 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
3769 020610 032701 010000      BIT #VLD,R1       ;CMR<12> SHOULD BE 1.
3770 020614 001010                    BNE 9$            ;PASS
3771                                ;
3772 020616 010537 047322      MOV R5,CA121     ;SAVE VALID STORE ADDRESS LOCATION
3773                                ;USED: CA<12:1>
3774 020622 006237 047322      ASR CA121        ;PREPARE CA121 FOR TYPEOUT
3775                                ;
020626 104000      ERROR          ;ERROR
020630 020626      .-2          ;-----
3776                                ;VALID BITS STORE TESTS
3777                                ;READING VALID STORE DATA SET A
3778                                ;THRU CMR<12> DID NOT RESULT IN 0.
3779 020632 047322      CA121          ;PRINT VALID STORE ADDRESS LOCATION
3780                                ;USED: CA<12:1>.
3781 020634 000000      0
3782                                ;
3783 020636 062705 000002      9$:  ADD #2,R5     ;NEXT VALID STORE LOCATION
3784 020642 062703 000002      ADD #2,R3
3785 020646 020527 100000      CMP R5,#100000   ;HAVE ALL HIGH CACHE ADDRESS LOCATIONS
3786                                ;BEEN DONE?
3787 020652 001334                    BNE 1$           ;NO
3788 020654 000240      10$:  NOP          ;END OF TEST
3789
3790
3791 020656 005237 001464      T127:  INC $TESTN ;UPDATE TEST ID

```

```

:*****
.SBTTL
.SBTTL TEST 127
.SBTTL

```

```

3792 .SBTTL VERIFY VALID DATA STORE ADDRESS LINES (CA<12:1>)
3793 .SBTTL
3794 .SBTTL
3795 ;*****
3796 020662 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
    020664 020674 40$ ;ERROR/LOOP ON TEST
    020666 070054 1$-40$+70000-14 ;TEST START LOCATION
    020670 000000 0 ;LOOP ON ERROR START LOCATION
    020672 070130 25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
    020674 012737 001015 177746 40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
    020702 004437 002352 JSR R4,RELCTH ;DISABLE CACHE
    020706 021124 10$+2 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
    ;ADDRESS OF START OF NEXT TEST

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

3797 020710 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
3798 020716 001407 BEQ 3$ ;YES
3799 020720 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A
3800 020726 032737 010000 177746 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3801 020734 001374 BNE -6
3802 020736 012737 000002 047332 3$: MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PA TERN
3803 020744 012702 040000 2$: MOV #40000,R2
3804 020750 012703 060000 MOV #60000,R3
3805 020754 063702 047332 ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
3806 020760 063703 047332 ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
3807 020764 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS VALID DATA STORE BITS TO BE
3808 ;WRITTEN TO CMR<12> ONLY DURING THE
3809 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3810 020772 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3811 021000 152737 000001 177750 BISB #TDAR,CMR ;TDAR WILL FORCE A 0 TO BE WRITTEN
3812 ;INTO VALID STORE WHEN A WRITE TO
3813 ;VALID STORE OCCURS.
3814 021006 005737 040000 TST 40000
3815 021012 005737 060000 TST 60000 ;WRITE 0 INTO VALID STORE LOCATION 0000.
3816 021016 142737 000001 177750 BICB #TDAR,CMR ;CLEARING TDAR WILL ALLOW A 1 TO BE
3817 ;WRITTEN INTO VALID STORE WHEN A WRITE
3818 ;TO VALID STORE OCCURS.
3819 021024 005712 TST (R2)
3820 021026 005713 TST (R3) ;WRITE 1 INTO VALID STORE LOCATION
3821 ;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
3822 021030 005737 060000 TST 60000 ;LOAD DATA FROM VALID DATA STORE LOCATION
3823 ;0000 INTO CMR<12>.
3824 021034 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
3825 021040 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
    021042 000240 NOP ;FOR LOOP ON ERROR

3826 021044 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
3827 021050 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3828 021056 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD READ 0.
3829 021062 001411 BEQ 9$ ;PASS
3830
3831 ; ROUTINE
3832 021064 013737 047332 047322 MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
3833 021072 006237 047322 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
    
```

```

3834 021076 104000 ERROR ;ERROR
;-----
021100 021076 .-2 ;VALID STOR ADDRESS VERIFICATION.
3835 ;VALID STORE LOCATION 0000 DID NOT
3836 ;READ AS A 0 INDICATING THAT IT WAS
3837 ;OVERWRITTEN WITH A 1. THIS SUGGESTS
3838 ;A BAD CA<12:1> VALID STORE ADDRESS
3839 ;LINE.
3840 ;PRINT VALID STORE ADDRESS FAILURE: CA<12:1>.
3841 021102 047322 CA121 ;NOTE THAT THE 1 IN THIS PATTERN
3842 ;WILL POINT TO THE ADDRESS LINE OF
3843 ;THAT BROUGHT OUT ERROR.
3844
3845 021104 000000 0
3846 ;;;
3847 021106 006337 047332 9$: ASL FLTPAT ;NEXT PATTERN
3848 021112 022737 020000 047332 CMP #20000,FLTPAT ;HAS VALID DATA STORE ADDRESS 4000 BEEN DONE?
3849 021120 001311 BNE 2$ ;NO
3850 021122 000240 10$: NOP ;END OF TEST
3851
3852
3853
3854 .SBTTL *
3855 .SBTTL
3856 .SBTTL * FLUSH CACHE INVALIDATION TEST - VALID STORE SET A
3857 .SBTTL
3858 .SBTTL *
3859
3860
3861
3862
3863 021124 005237 001464 T130: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 130
.SBTTL
.SBTTL VERIFY THAT ALL LOW CACHE VALID STORE SET A ADDRESS LOCATIONS
3864 .SBTTL WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
3865 .SBTTL (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 0000-3777)
3866 .SBTTL
3867 .SBTTL
3868 ;*****
3869
3870 021130 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
021132 021142 40$ ;TEST START LOCATION
021134 070074 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
021136 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
021140 070130 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
021142 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
021150 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
021154 021402 10$+2 ;ADDRESS OF START OF NEXT TEST

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

```

```

3872 021156 012705 060000      MOV #60000,R5      ;;ADDRESS 60000 INTO R5
3873 021162 012703 040000      MOV #40000,R3      ;;ADDRESS 40000 INTO R3
3874 021166 032737 020000 177746  BIT #VSIU,CCR      ;;IS SET A BEING USED?
3875 021174 001407              BEQ 3$             ;;YES
3876 021176 052737 000400 177746  BIS #FC,CCR        ;;NO; FLUSH CACHE FOR SET A
3877 021204 C32737 010000 177746  BIT #VCIP,CCR      ;;WAIT TILL FLUSH COMPLETE
3878 021212 001374              BNE .-6
3879 021214 012737 000015 177746  3$: MOV #15,CCR      ;;NO UCB SO AS TO WRITE ENABLE VALID STORE
3880 021222 112737 000002 177750  MOVB #HODO,CMR    ;;HODO ALLOWS VALID STORE SET A TO
3881                          ;;BE WRITTEN TO CMR<12> ONLY DURING
3882                          ;;THE DESTINATION MEMORY ACCESS.
3883 021230 005713              4$: TST (R3)
3884 021232 005715              TST (R5)          ;;WRITE A 1 INTO VALID STORE ADDRESS
3885                          ;;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3886 021234 062705 000002      ADD #2,R5          ;;NEXT VALID STORE LOCATION
3887 021240 062703 000002      ADD #2,R3
3888 021244 020527 070000      CMP R5,#70000     ;;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
3889 021250 001367              BNE 4$            ;;NO
3890 021252 052737 000400 177746  1$: BIS #FC,CCR    ;;FLUSH CACHE TO SELECT SET B AND
3891                          ;;INVALIDATE SET A
3892 021260 032737 010000 177746  BIT #VCIP,CCR      ;;WAIT TILL FLUSH COMPLETE
3893 021266 001374              BNE .-6
3894 021270 052737 000400 177746  BIS #FC,CCR        ;;FLUSH TO SELECT SET A AGAIN
3895 021276 032737 010000 177746  BIT #VCIP,CCR      ;;WAIT
3896 021304 001374
3897      021306 000240          25$: NOP          ;;INSTRUCTION 'JMP 1$' PLACED HERE
3898      021310 000240          NOP              ;;FOR LOOP ON ERROR
3899 021312 012705 060000      2$: MOV #60000,R5  ;;ADDRESS 60000 INTO R5
3900 021316 005715          TST (R5)          ;;WRITE VALID STORE DATA INTO CMR<12>
3901                          ;;FROM ADDRESS LOCATION SPECIFIED BY
3902                          ;;R5'S BITS 12-1.
3903 021320 013701 177750      MOV CMR,R1        ;;SAVE CMR DATA
3904 021324 032701 010000      BIT #VLD,R1      ;;CMR<12> SHOULD BE 0
3905 021330 001416              BEQ 9$            ;;PASS
3906 021332 105037 177750      CLRB CMR         ;;DISABLE MAINT. MODE
3907 021336 012737 001015 177746  MOV #OFF,CCR      ;;DISABLE CACHE
3908
3909 021344 010537 047330      MOV R5,CNT121    ;;SAVE VALID STORE FLUSH ADDRESS LOCATION
3910                          ;;USED: CNT<12:1>
3911 021350 006237 047330      ASR CNT121       ;;PREPARE CNT121 FOR TYPEOUT
3912      021354 104000          ERROR              ;;ERROR
3913      021356 021354          .-2              ;;-----
3914                          ;;FLUSH CACHE INVALID TEST-SET A
3915                          ;;READING VALID STORE LOCATION FROM SET A THRU CMR<12>
3916                          ;;DID NOT RESULT IN A ZERO,INDICATING THAT
3917                          ;;THE CACHE FLUSH DID NOT INVALIDATE THIS
3918 021360 047330          CNT121          ;;LOCATION.
3919                          ;;PRINT VALID STORE FLUSH ADDRESS LOCATION
3920 021362 000000          C              ;;IN ERROR: CNT<12:1>.
3921 021364 000405          BR 10$         ;;IF ERROR, END TEST
3922

```

...

```

3923 021366 062705 000002          9$:  ADD #2,R5          ;NEXT VALID STORE LOCATION
3924 021372 020527 070000          CMP R5,#70000        ;HAVE ALL LOW CACHE ADDRESS LOCATIONS
3925                                     ;BEEN DONE?
3926 021376 001347          BNE 2$              ;NO
3927 021400 000240          10$: NOP              ;END OF TEST
3928
3929
3930
3931 021402 005237 001464          T131: INC $TESTN      ;UPDATE TEST ID
;:*****
.SBTTL
.SBTTL TEST 131
.SBTTL
.SBTTL VERIFY THAT ALL HI CACHE VALID STORE SET A ADDRESS LOCATIONS
3932 .SBTTL WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
3933 .SBTTL (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 4000-7777)
3934 .SBTTL
3935 ;:*****
3936
3937
3938
3939 021406 000004          SCPCND              ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                40$
                                ;TEST START LOCATION
                                1$-40$+60000-14
                                ;LOOP ON ERROR START LOCATION
                                0
                                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                25$-40$+60000-14
                                ;LOOP ON ERROR END LOCATION
                                40$: MOV #OFF,CCR        ;DISABLE CACHE
                                JSR R4,RELCTL      ;LOCATE TEST CODE TO LOW CACHE SPACE
                                10$+2            ;ADDRESS OF START OF NEXT TEST

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

3940 021434 012705 070000          MOV #70000,R5       ;;ADDRESS 70000 INTO R5
3941 021440 012703 050000          MOV #50000,R3       ;ADDRESS 50000 INTO R3
3942 021444 032737 020000 177746   BIT #VSIU,CCR       ;IS SET A BEING USED?
3943 021452 001407          BEQ 3$              ;YES
3944 021454 052737 000400 177746   BIS #FC,CCR         ;NO; FLUSH CACHE FOR SET A
3945 021462 032737 010000 177746   BIT #VCIP,CCR       ;WAIT TILL FLUSH COMPLETE
3946 021470 001374          BNE -6
3947 021472 012737 000015 177746   3$: MOV #15,CCR      ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3948 021500 112737 000002 177750   MOVB #HODO,CMR     ;HODO ALLOWS VALID STORE SET A TO
3949                                     ;BE WRITTEN TO CMR<12> ONLY DURING
3950                                     ;THE DESTINATION MEMORY ACCESS.
3951 021506 005713          4$: TST (R3)
3952 021510 005715          TST (R5)           ;WRITE A 1 INTO VALID STORE ADDRESS
3953                                     ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3954 021512 062705 000002          ADD #2,R5          ;NEXT VALID STORE LOCATION
3955 021516 062703 000002          ADD #2,R3
3956 021522 020527 100000          CMP R5,#100000     ;HAVE ALL HI CACHE LOCATIONS BEEN DONE?
3957 021526 001367          BNE 4$            ;NO
3958 021530 052737 000400 177746   1$: BIS #FC,CCR     ;FLUSH CACHE TO SELECT SET B AND
3959                                     ;INVALIDATE SET A
3960 021536 032737 010000 177746   BIT #VCIP,CCR       ;WAIT TILL FLUSH COMPLETE
3961 021544 001374          BNE -6
3962 021546 052737 000400 177746   BIS #FC,CCR         ;FLUSH TO SELECT SET A AGAIN
3963 021554 032737 010000 177746   BIT #VCIP,CCR       ;WAIT

```

```

3964 021562 001374          BNE  .-6
3965          021564 000240      25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          021566 000240          NOP          ;FOR LOOP ON ERROR

3966
3967 021570 012705 070000      2$:  MOV #70000,R5      ;ADDRESS 70000 INTO R5
3968 021574 005715          TST (R5)      ;WRITE VALID STORE DATA INTO (MR<12>
3969          ;FROM ADDRESS LOCATION SPECIFIED BY
3970          ;R5'S BITS 12-1.
3971 021576 013701 177750      MOV CMR,R1      ;SAVE CMR DATA
3972 021602 032701 010000      BIT #VLD,R1     ;CMR<12> SHOULD BE 0
3973 021606 001416          BEQ 9$         ;PASS
3974 021610 105037 177750      CLRB CMR        ;DISABLE MAINT. MODE
3975 021614 012737 001015 177746 MOV #OFF,CCR     ;DISABLE CACHE
3976
3977 021622 010537 047330      MOV R5,CNT121   ;SAVE VALID STORE FLUSH ADDRESS LOCATION
3978          ;USED: CNT<12:1>
3979 021626 006237 047330      ASR CNT121      ;PREPARE CNT121 FOR TYPEOUT
3980
          021632 104000          ERROR          ;ERROR
          ;-----
          021634 021632          .-2

3981          ;FLUSH CACHE TEST-SET A
3982          ;READING VALID STORE LOCATION FROM SET A THRU CMR<12
3983          ;DID NOT RESULT IN A ZERO,INDICATING THAT
3984          ;THE CACHE FLUSH DID NOT INVALIDATE THIS
3985          ;LOCATION.
3986 021636 047330          CNT*21      ;PRINT VALID STORE FLUSH ADDRESS LOCATION
3987          ;IN ERROR: CNT<12:1>.
3988          0
3989 021642 000405          BR 10$         ;IF ERROR, END TEST
3990          ;:
3991 021644 062705 000002      9$:  ADD #2,R5      ;NEXT VALID STORE LOCATION
3992 021650 020527 100000      CMP R5,#100000 ;HAVE ALL HI CACHE ADDRESS LOCATIONS
3993          ;BEEN DONE?
3994 021654 001347          BNE 2$         ;NO
3995 021656 000240      10$:  NOP          ;END OF TEST
3996
3997
3998
3999
4000
4001          .SBTTL *
4002          .SBTTL
4003          .SBTTL *
4004          .SBTTL
4005          .SBTTL *
4006
4007
4008
4009 021660 005237 001464      1132: INC $TESTN      ;UPDATE TEST ID
          ;*****
          .SBTTL
          .SBTTL TEST 132
          .SBTTL
          .SBTTL WRITE AND READ 0'S TO ALL LOW CACHE VALID
4010

```

```
4011 .SBTTL BIT STORE ADDRESS LOCATIONS- SET B
4012 .SBTTL (VALID STORE LOCATIONS 0000 TO 3777)
4013 .SBTTL
4014 :*****
4015 021664 000004 SPCOND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
021666 021676 40$ ;TEST START LOCATION
021670 070052 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
021672 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
021674 070100 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
021676 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
021704 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
021710 022076 10$+2 ;ADDRESS OF START OF NEXT TEST
```

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

```
4016 021712 032737 020000 177746 BIT #VSIU,CCR ;IS SET B BEING USED?
4017 021720 001007 BNE 3$ ;YES
4018 021722 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B
4019 021730 032737 010000 177746 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
4020 021736 001374 BNE -6
4021 021740 012737 177777 177752 3$: MOV #-1,CHR ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
4022 ;REGISTERS, SINCE TDAR WILL BE USED
4023 021746 112737 000374 177751 MOVB #374,CMR+1
4024 021754 012705 060000 MOV #60000,R5 ;: ADDRESS 60000 INTO R5
4025 021760 012703 040000 MOV #40000,R3 ;: ADDRESS 40000 INTO R3
4026 021764 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
4027 021772 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
4028
4029
4030
4031
4032
4033 022000 005713 TST (R3)
4034 022002 005715 TST (R5) ;WRITE A 0 INTO VALID STORE ADDRESS
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
4035 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
;FROM VALID STORE ADDRESS LOCATION
;JUST WRITTEN INTO.
4036 022004 005715 ;SAVE CMR DATA
4037
4038
4039 022006 013701 177750 MOV CMR,R1
4040 022012 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
022014 000240 NOP ;FOR LOOP ON ERROR
4041 022016 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
4042 022022 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4043 022030 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 0.
4044 022034 001410 BEQ 9$ ;PASS
4045
4046 022036 010537 047322 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
4047 ;USED: CA<12:1>
4048 022042 006237 047322 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
4049 022046 104000 ERROR ;ERROR
022050 022046 .- ;-----
```



```

022214 000240          25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
022216 000240          NOP          ;FOR LOOP ON ERROR

4091 022220 105037 177750          CLRB CMR          ;DISABLE MAINT. MODE
4092 022224 012737 001015 177746  MOV #OFF,CCR      ;DISABLE CACHE
4093 022232 032701 010000          BIT #VLD,R1       ;CMR<12> SHOULD BE 1.
4094 022236 001010          BNE 9$           ;PASS
4095
4096 022240 010537 047322          MOV R5,CA121      ;SAVE VALID STORE ADDRESS LOCATION
4097                                ;USED: CA<12:1>
4098 022244 006237 047322          ASR CA121         ;PREPARE CA121 FOR TYPEOUT
4099
022250 104000          ERROR          ;ERROR
                                ;-----
022252 022250          .-2
4100                                ;VALID BITS STORE TESTS - SET B
4101                                ;READING VALID STORE DATA SET B
4102                                ;THRU CMR<12> DID NOT RESULT IN 1.
4103 022254 047322          CA121            ;PRINT VALID STORE ADDRESS LOCATION
4104                                ;USED: CA<12:1>.
4105 022256 000000          0
4106                                ;:
4107 022260 062705 000002          9$:  ADD #2,R5      ;NEXT VALID STORE LOCATION
4108 022264 062703 000002          ADD #2,R3
4109 022270 020527 070000          CMP R5,#70000    ;HAVE ALL LOW CACHE ADDRESS LOCATIONS
4110                                ;BEEN DONE?
4111 022274 001334          BNE 1$           ;NO
4112 022276 000240          10$: NOP          ;END OF TEST
4113
4114
4115
4116 022300 005237 001464          T134:          INC $TESTN          ;UPDATE TEST ID
                                ;:*****
                                ;.SBTTL
                                ;.SBTTL TEST 134
                                ;.SBTTL
                                ;.SBTTL WRITE AND READ 0'S TO ALL HIGH CACHE VALID
4117                                ;.SBTTL BIT STORE ADDRESS LOCATIONS- SET B
4118                                ;.SBTTL (VALID STORE LOCATIONS 4000 TO 7777)
4119                                ;.SBTTL
4120                                ;.SBTTL
4121                                ;:*****
4122 022304 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
022306 022316          40$          ;LOOP ON ERROR START LOCATION
022310 060052          1$-40$+60000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
022312 000000          0          ;LOOP ON ERROR END LOCATION
022314 060100          25$-40$+60000-14 ;DISABLE CACHE
022316 012737 001015 177746  40$: MOV #OFF,CCR  ;LOCATE TEST CODE TO LOW CACHE SPACE
022324 004437 002324          JSR R4,RELCTL    ;ADDRESS OF START OF NEXT TEST
022330 022516          10$+2
                                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                ;ARE RELOCATED TO LOW CACHE SPACE

4123 022332 032737 020000 177746  BIT #VSIU,CCR     ;IS SET B BEING USED?
4124 022340 001007          BNE 3$           ;YES
4125 022342 052737 000400 177746  BIS #FC,CCR       ;NO: FLUSH CACHE FOR SET B

```

```

4126 022350 032737 010000 177746          BIT #VCIP,CCR          ;WAIT TILL FLUSH COMPLETE
4127 022356 001374                      BNE .-6
4128 022360 012737 177777 177752          3$: MOV #-1,CHR          ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
4129                      ;REGISTERS, SINCE TDAR WILL BE USED
4130 022366 112737 000374 177751          MOVB #374,CMR+1
4131 022374 012705 070000          MOV #70000,R5          ;:ADDRESS 70000 INTO R5
4132 022400 012703 050000          MOV #50000,R3          ;:ADDRESS 50000 INTO R3
4133 022404 012737 000015 177746          1$: MOV #15,CCR          ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
4134 022412 112737 000003 177750          MOVB #HODO+TDAR,CMR   ;:HODO ALLOWS VALID STORE SET B TO
4135                      ;BE WRITTEN TO CMR<12> ONLY DURING
4136                      ;THE DESTINATION MEMORY ACCESS.
4137                      ;TDAR WILL FORCE A 0 TO BE WRITTEN
4138                      ;INTO VALID STORE WHEN A WRITE TO
4139                      ;VALID STORE OCCURS
4140 022420 005713                      TST (R3)
4141 022422 005715                      TST (R5)          ;:WRITE A 0 INTO VALID STORE ADDRESS
4142                      ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
4143 022424 005715                      TST (R5)          ;:WRITE VALID STORE DATA INTO CMR<12>
4144                      ;FROM VALID STORE ADDRESS LOCATION
4145                      ;JUST WRITTEN INTO.
4146 022426 013701 177750          MOV CMR,R1          ;:SAVE CMR DATA
4147                      ;
          022432 000240          25$: NOP          ;:INSTRUCTION 'JMP 1$' PLACED HERE
          022434 000240          NOP          ;:FOR LOOP ON ERROR
4148 022436 105037 177750          CLRB CMR          ;:DISABLE MAINT. MODE
4149 022442 012737 001015 177746          MOV #OFF,CCR        ;:DISABLE CACHE
4150 022450 032701 010000          BIT #VLD,R1        ;:CMR<12> SHOULD BE 0.
4151 022454 001410          BEQ 9$            ;:PASS
4152                      ;
4153 022456 010537 047322          MOV R5,CA121       ;:SAVE VALID STORE ADDRESS LOCATION
4154                      ;USED: CA<12:1>
4155 022462 006237 047322          ASR CA121          ;:PREPARE CA121 FOR TYPEOUT
4156                      ;
          022466 104000          ERROR              ;:ERROR
          ;-----
          022470 022466          .-2
4157                      ;:VALID BITS STORE TESTS - SET B
4158                      ;:READING VALID STORE DATA SET B
4159                      ;:THRU CMR<12> DID NOT RESULT IN 0.
4160 022472 047322          CA121             ;:PRINT VALID STORE ADDRESS LOCATION
4161                      ;USED: CA<12:1>.
4162 022474 000000          0
4163                      ;
          022476 062705 000002          ::: 9$: ADD #2,R5          ;:NEXT VALID STORE LOCATION
4164 022502 062703 000002          ADD #2,R3
4165 022506 020527 100000          CMP R5,#100000     ;:HAVE ALL HIGH CACHE ADDRESS LOCATIONS
4166                      ;BEEN DONE?
4167                      ;NO
4168 022512 001334          BNE 1$            ;:NO
4169 022514 000240          10$: NOP          ;:END OF TEST
4170                      ;
4171                      ;
4172 022516 005237 001464          T135: INC $TESTN    ;:UPDATE TEST ID

```

```

:*****
.SBTTL
.SBTTL TEST 135
.SBTTL

```

```

4173 .SBTTL WRITE AND READ 1'S TO ALL HIGH CACHE VALID
4174 .SBTTL BIT STORE ADDRESS LOCATIONS- SET B
4175 .SBTTL (VALID STORE LOCATIONS 4000 TO 7777)
4176 .SBTTL
4177 :*****
4178 022522 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
    ;ERROR/LOOP ON TEST
    022524 022534 40$ ;TEST START LOCATION
    022526 060036 1$-40$+60000-14 ;LOOP ON ERROR START LOCATION
    022530 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
    022532 060064 25$-40$+60000-14 ;LOOP ON ERROR END LOCATION
    022534 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
    022542 004437 002324 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
    022546 022720 10$+2 ;ADDRESS OF START OF NEXT TEST
    
```

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

```

4179 022550 032737 020000 177746 BIT #VSIU,CCR ;IS SET B BEING USED?
4180 022556 001007 BNE 3$ ;YES
4181 022560 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B
4182 022566 032737 010000 177746 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
4183 022574 001374 BNE -6
4184 022576 012705 070000 3$: MOV #70000,R5 ;:ADDRESS 70000 INTO R5
4185 022602 012703 050000 MOV #50000,R3 ;:ADDRESS 50000 INTO R3
4186 022606 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
4187 022614 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS VALID STORE SET B TO
4188 ;BE WRITTEN TO CMR<12> ONLY DURING
4189 ;THE DESTINATION MEMORY ACCESS.
4190 022622 005713 TST (R3)
4191 022624 005715 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS
4192 ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
4193 022626 005715 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
4194 ;FROM VALID STORE ADDRESS LOCATION
4195 ;JUST WRITTEN INTO.
4196 022630 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
4197 ;INSTRUCTION 'JMP 1$' PLACED HFRE
    022634 000240 25$: NOP ;FOR LOOP ON ERROR
    022636 000240 NOP
4198 022640 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
4199 022644 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4200 022652 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.
4201 022656 001010 BNE 9$ ;PASS
4202
4203 022660 010537 047322 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
4204 ;USED: CA<12:1>
4205 022664 006237 047322 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
4206 ;ERROR
    022670 104000 ERROR ;ERROR
    ;-----
    022672 022670 -2 ;VALID STORE BIT TEST- SET B
4207 ;READING VALID STORE DATA SET B
4208 ;THRU CMR<12> DID NOT RESULT IN 0.
4209 ;PRINT VALID STORE ADDRESS LOCATION
4210 022674 047322 CA121 ;USED: CA<12:1>.
4211
    
```

```

4212 022676 000000          0
4213          :::
4214 022700 062705 000002    9$:  ADD #2,R5          ;NEXT VALID STORE LOCATION
4215 022704 062703 000002    ADD #2,R3
4216 022710 020527 100000    CMP R5,#100000      ;HAVE ALL HIGH CACHE ADDRESS LOCATIONS
4217          ;BEEN DONE?
4218 022714 001334          BNE 1$              ;NO
4219 022716 000240          10$:  NOP                ;END OF TEST
4220
4221
4222 022720 005237 001464    T136:  INC $TESTN      ;UPDATE TEST ID
;:*****
.SBTTL
.SBTTL TEST 136
.SBTTL
.SBTTL VERIFY VALID DATA STORE ADDRESS LINES (CA<12:1>) - SET B
.SBTTL
.SBTTL
;:*****
4223
4224
4225
4226
4227 022724 000004          SCPCND              ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
022726 022736          40$
022730 070054          1$-40$+70000-14   ;TEST START LOCATION
022732 000000          0                  ;LOOP ON ERROR START LOCATION
022734 070130          25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
022736 012737 001015 177746 40$:  MOV #OFF,CCR      ;LOOP ON ERROR END LOCATION
022744 004437 002352          JSR R4,RELCTH      ;DISABLE CACHE
022750 023166          10$+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

4228 022752 032737 020000 177746    BIT #VSIU,CCR      ;IS SET B BEING USED?
4229 022760 001007          BNE 3$              ;YES
4230 022762 052737 000400 177746    BIS #FC,CCR        ;NO; FLUSH CACHE FOR SET B
4231 022770 032737 010000 177746    BIT #VCIP,CCR      ;WAIT TILL FLUSH COMPLETE
4232 022776 001374          BNE .-6
4233 023000 012737 000002 047332    3$:  MOV #2,FLTPAT    ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
4234 023006 012702 040000    2$:  MOV #40000,R2
4235 023012 012703 060000    MOV #60000,R3
4236 023016 063702 047332    ADD FLTPAT,R2
4237 023022 063703 047332    ADD FLTPAT,R3
4238 023026 112737 000002 177750    1$:  MOVB #HODO,CMR
4239
4240
4241 023034 012737 000015 177746    MOV #15,CCR
4242 023042 152737 000001 177750    BISB #TDAR,CMR
4243
4244
4245 023050 005737 040000          TST 40000
4246 023054 005737 060000          TST 60000
4247 023060 142737 000001 177750    BICB #TDAR,CMR
4248
4249
4250 023066 005712          TST (R2)
4251 023070 005713          TST (R3)
4252
;WRITE 0 INTO VALID STORE LOCATION 0000.
;CLEARING TDAR WILL ALLOW A 1 TO BE
;WRITTEN INTO VALID STORE WHEN A WRITE
;TO VALID STORE OCCURS.
;WRITE 1 INTO VALID STORE LOCATION
;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.

```

```

4253 023072 005737 060000      TST 60000      ;LOAD DATA FROM VALID DATA STORE LOCATION
4254                               ;0000 INTO CMR<12>.
4255 023076 013701 177750      MOV CMR,R1     ;SAVE CMR DATA
4256                               ;INSTRUCTION 'JMP 1$' PLACED HERE
      023102 000240      25$:  NOP      ;FOR LOOP ON ERROR
      023104 000240      NOP
4257 023106 105037 177750      CLRB CMR      ;DISABLE MAINT. MODE
4258 023112 012737 001015 177746  MOV #OFF,CCR  ;DISABLE CACHE
4259 023120 032701 010000      BIT #VLD,R1   ;CMR<12> SHOULD READ 0.
4260 023124 001411      BEQ 9$        ;PASS
4261
4262                               ; ROUTINE
4263 023126 013737 047332 047322  MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
4264 023134 006237 047322      ASR CA121     ;PREPARE CA121 FOR TYPEOUT
4265
      023140 104000      ERROR      ;ERROR
      023142 023140      .-2          ;-----
4266                               ;VALID STORE ADDRESS VERIFICATION- SET B
4267                               ;VALID STORE LOCATION 0000 DID NOT
4268                               ;READ AS A 0 INDICATING THAT IT WAS
4269                               ;OVERWRITTEN WITH A 1. THIS SUGGESTS
4270                               ;A BAD CA<12:1> VALID STORE ADDRESS
4271                               ;LINE.
4272 023144 047322      CA121      ;PRINT VALID STORE ADDRESS FAILURE: CA<12:1>.
4273                               ;NOTE THAT THE 1 IN THIS PATTERN
4274                               ;WILL POINT TO THE ADDRESS LINE
4275                               ;THAT BROUGHT OUT ERROR.
4276 023146 000000      0
4277                               ;:
4278 023150 006337 047332 047332  9$:  ASL FLTPAT ;NEXT PATTERN
4279 023154 022737 020000 047332  CMP #20000,FLTPAT ;HAS VALID DATA STORE ADDRESS 4000 BEEN DONE?
4280 023162 001311      BNE 2$        ;NO
4281 023164 000240      10$:  NOP      ;END OF TEST
4282
4283
4284
4285      .SBTTL *
4286      .SBTTL
4287      .SBTTL *      FLUSH CACHE INVALIDATION TEST- VALID STORE SET B
4288      .SBTTL
4289      .SBTTL *
4290
4291
4292
4293
4294 023166 005237 001464      T137:      INC $TESTN      ;UPDATE TEST ID
      ;*****
      .SBTTL
      .SBTTL TEST 137
      .SBTTL
      .SBTTL VERIFY THAT ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS
      .SBTTL WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
      .SBTTL (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 0000-3777)
      .SBTTL
      ;*****
4295
4296
4297
4298
4299

```

```

4300
4301 023172 000004          SCPCND          ; SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;:RROR/LOOP ON TEST
                                ;:EST 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$
                                40$
                                1$-40$+70000-14
                                0
                                25$-40$+70000-14
                                40$: MOV #OFF,CCR
                                JSR R4,RELCTH
                                10$+2

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

4302 023220 012705 060000          MOV #60000,R5          ;; ADDRESS 60000 INTO R5
4303 023224 012703 040000          MOV #40000,R3          ;; ADDRESS 40000 INTO R3
4304 023230 032737 020000 177746   BIT #VSIU,CCR          ;: IS SET B BEING USED?
4305 023236 001007                BNE 3$                 ;: YES
4306 023240 052737 000400 177746   BIS #FC,CCR            ;: NO; FLUSH CACHE FOR SET B
4307 023246 032737 010000 177746   BIT #VCIP,CCR          ;: WAIT TILL FLUSH COMPLETE
4308 023254 001374                BNE .-6
4309 023256 012737 000015 177746   3$: MOV #15,CCR          ;: NO UCB SO AS TO WRITE ENABLE VALID STORE
4310 023264 112737 000002 177750   MOVB #HODO,CMR        ;: HODO ALLOWS VALID STORE SET B TO
4311                                ;: BE WRITTEN TO CMR<12> ONLY DURING
4312                                ;: THE DESTINATION MEMORY ACCESS.
4313 023272 005713                4$: TST (R3)
4314 023274 005715                TST (R5)
4315                                ;: WRITE A 1 INTO VALID STORE ADDRESS
4316 023276 062705 000002          ADD #2,R5              ;: LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
4317 023302 062703 000002          ADD #2,R3              ;: NEXT VALID STORE LOCATION
4318 023306 020527 070000          CMP R5,#70000          ;: HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
4319 023312 001367                BNE 4$                 ;: NO
4320 023314 052737 000400 177746   1$: BIS #FC,CCR          ;: FLUSH CACHE TO SELECT SET A AND
4321                                ;: INVALIDATE SET B
4322 023322 032737 010000 177746   BIT #VCIP,CCR          ;: WAIT TILL FLUSH COMPLETE
4323 023330 001374                BNE .-6
4324 023332 052737 000400 177746   BIS #FC,CCR            ;: FLUSH TO SELECT SET B AGAIN
4325 023340 032737 010000 177746   BIT #VCIP,CCR          ;: WAIT
4326 023346 001374                BNE .-6
4327                                25$: NUP
                                NOP
                                ;: INSTRUCTION 'JMP 1$' PLACED HERE
                                ;: FOR LOOP ON ERROR

4328
4329 023354 012705 060000          2$: MOV #60000,R5          ;: ADDRESS 60000 INTO R5
4330 023360 005715                TST (R5)              ;: WRITE VALID STORE DATA INTO CMR<12>
4331                                ;: FROM ADDRESS LOCATION SPECIFIED BY
4332                                ;: R5'S BITS 12-1.
4333 023362 013701 177750          MOV CMR,R1             ;: SAVE CMR DATA
4334 023366 032701 010000          BIT #VLD,R1           ;: CMR<12> SHOULD BE 0
4335 023372 001416                BEQ 9$                 ;: PASS
4336 023374 105037 177750          CLRB CMR              ;: DISABLE MAINT MODE
4337 023400 012737 001015 177746   MOV #OFF,CCR          ;: DISABLE CACHE
4338 023406 010537 047330          MOV R5,CNT121         ;: SAVE VALID STORE FLUSH ADDRESS LOCATION
4339                                ;: USED: CNT<12:1>
4340 023412 006237 047330          ASR CNT121            ;: PREPARE CN:121 FOR TYPEOUT
4341

```

```

023416 104000 ERROR ;ERROR
;-----
023420 023416 .-2 ;FLUSH CACHE INVALID TEST-SET B
4342 ;READING VALID STORE LOCATION FROM SET B THRU CMR<12>
4343 ;DID NOT RESULT IN A ZERO,INDICATING THAT
4344 ;THE CACHE FLUSH DID NOT INVALIDATE THIS
4345 ; LOCATION.
4346 ;PRINT VALID STORE FLUSH ADDRESS LOCATION
4347 023422 047330 CNT121 ;IN ERROR: CNT<12:1>.
4348 0 ;IF ERROR, END TEST
4349 023424 000000 0 ;NEXT VALID STORE LOCATION
4350 023426 000405 BR 10$ ;HAVE ALL LOW CACHE ADDRESS LOCATIONS
4351 023430 062705 000002 9$: ADD #2,R5 ;BEEN DONE?
4352 023434 020527 070000 CMP R5,#70000 ;NO
4353 ;END OF TEST
4354 023440 001347 10$: BNE 2$
4355 023442 000240 NOP
4356
4357
4358
4359 023444 005237 001464 T140: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 140
.SBTTL
.SBTTL VERIFY THAT ALL HI CACHE VALID STORE SET B ADDRESS LOCATIONS
4360 .SBTTL WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
4361 .SBTTL (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 4000-7777)
4362 .SBTTL
4363 ;*****
4364
4365
4366 023450 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
023452 023462 40$ ;TEST START LOCATION
023454 060074 1$-40$+60000-14 ;LOOP ON ERROR START LOCATION
023456 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
023460 060130 25$-40$+60000-14 ;LOOP ON ERROR END LOCATION
023462 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
023470 004437 002324 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
023474 023722 10$+2 ;ADDRESS OF START OF NEXT TEST

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

4367 023476 012705 070000 MOV #70000,R5 ;:ADDRESS 70000 INTO R5
4368 023502 012703 050000 MOV #50000,R3 ;:ADDRESS 50000 INTO R3
4369 023506 032737 020000 177746 BIT #VSIU,CCR ;:IS SET B BEING USED?
4370 023514 001007 BNE 3$ ;:YES
4371 023516 052737 000400 177746 BIS #FC,CCR ;:NO; FLUSH CACHE FOR SET B
4372 023524 032737 010000 177746 BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
4373 023532 001374 BNE .-6
4374 023534 012737 000015 177746 3$: MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
4375 023542 112737 000002 177750 MOVB #HODO,CMR ;:HODO ALLOWS VALID STORE SET B TO
4376 ;:BE WRITTEN TO CMR<12> ONLY DURING
4377 ;:THE DESTINATION MEMORY ACCESS.
4378 023550 005713 4$: TST (R3)
4379 023552 005715 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS

```



```

4380                                     ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
4381 023554 062705 000002             ADD #2,R5                ;NEXT VALID STORE LOCATION
4382 023560 062703 000002             ADD #2,R3
4383 023564 020527 100000             CMP R5,#100000         ;HAVE ALL HI CACHE LOCATIONS BEEN DONE?
4384 023570 001367                    BNE 4$                 ;NO
4385 023572 052737 000400 177746     1$: BIS #FC,CCR          ;FLUSH CACHE TO SELECT SET A AND
4386                                     ;INVALIDATE SET B
4387 023600 032737 010000 177746     BIT #VCIP,CCR          ;WAIT TILL FLUSH COMPLETE
4388 023606 001374                    BNE .-6
4389 023610 052737 000400 177746     BIS #FC,CCR           ;FLUSH TO SELECT SET B AGAIN
4390 023616 032737 010000 177746     BIT #VCIP,CCR          ;WAIT
4391 023624 001374                    BNE .-6
4392                                     ;INSTRUCTION 'JMP 1$' PLACED HERE
                                25$: NOP                ;FOR LOOP ON ERROR
                                NOP
4393
4394 023632 012705 070000             MOV #70000,R5          ;ADDRESS 70000 INTO R5
4395 023636 005715                    TST (R5)               ;WRITE VALID STORE DATA INTO CMR<12>
4396                                     ;FROM ADDRESS LOCATION SPECIFIED BY
4397                                     ;R5'S BITS 12-1.
4398 023640 013701 177750             MOV CMR,R1            ;SAVE CMR DATA
4399 023644 032701 010000             BIT #VLD,R1           ;CMR<12> SHOULD BE 0
4400 023650 001416                    BEQ 9$                 ;PASS
4401 023652 105037 177750             CLR# CMR              ;DISABLE MAINT MODE
4402 023656 012737 001015 177746     MOV #OFF,CCR          ;DISABLE CACHE
4403
4404 023664 010537 047330             MOV R5,CNT121         ;SAVE VALID STORE FLUSH ADDRESS LOCATION
4405                                     ;USED: CNT<12:1>
4406 023670 006237 047330             ASR CNT121            ;PREPARE CNT121 FOR TYPEOUT
4407
                                ERROR                    ;ERROR
                                .-2                      ;-----
4408                                     ;FLUSH CACHE INVALID TEST-SET B
4409                                     ;READING VALID STORE LOCATION FROM SET B THRU CMR<12>
4410                                     ;DID NOT RESULT IN A ZERO,INDICATING THAT
4411                                     ;THE CACHE FLUSH DID NOT INVALIDATE THIS
4412                                     ;LOCATION.
4413 023700 047330                     CNT121                ;PRINT VALID STORE FLUSH ADDRESS LOCATION
4414                                     ;IN ERROR: CNT<12:1>.
4415 023702 000000                     0
4416 023704 000405                    BR 10$                 ;IF ERROR,END TEST
4417 023706 062705 000002             9$: ADD #2,R5          ;NEXT VALID STORE LOCATION
4418 023712 020527 100000             CMP R5,#100000        ;HAVE ALL HI CACHE ADDRESS LOCATIONS
4419                                     ;BEEN DONE?
4420 023716 001347                    BNE 2$                 ;NO
4421 023720 000240                    10$: NOP                ;END OF TEST
4422
4423
4424
4425
4426
4427 .SBTTL *
4428 .SBTTL
4429 .SBTTL *
4430 .SBTTL

```

WRITE CONTROL LOGIC TESTS

4431 .SBTTL *
 4432
 4433
 4434
 4435 023722 005237 001464

T141: INC \$TESTN ;UPDATE TEST ID
 ;*****

4436
 4437
 4438
 4439
 4440
 4441
 4442 023726 000004
 023730 023740
 023732 070000
 023734 000000
 023736 070064
 023740 012737 001015 177746
 023746 004437 002352
 023752 024146

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

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 40\$;ERROR/LOOP ON TEST
 1\$-40\$+70000-14 ;TEST START LOCATION
 0 ;LOOP ON ERROR START LOCATION
 25\$-40\$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
 40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
 JSR R4,RELCTH ;DISABLE CACHE
 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

4443 023754 005037 060000
 4444 023760 005037 000000
 4445 023764 012700 177777
 4446 023770 012701 060000
 4447 023774 112737 000002 177750
 4448
 4449
 4450 024002 012737 000015 177746
 4451 024010 005737 060000
 4452 024014 005737 000000
 4453
 4454
 4455 024020 010011
 4456
 4457 024022 005711
 4458
 4459 024024 013702 177754
 4460 024030 010011
 4461
 4462
 4463 024032 005711
 4464
 4465 024034 013703 177754
 4466
 024040 000240
 024042 000240
 4467 024044 105037 177750
 4468 024050 012737 001015 177746

1\$: CLR 60000 ;0'S TO MAIN MEMORY LOCATION
 CLR 2#0 ;CLEAR LOCATION 0
 MOV #-1,R0 ;ALL 1'S TO R0
 MOV #60000,R1 ;ADDRESS 60000 TO R1
 MOVB #HODO,CMR ;ALLOWS CACHE UPDATES & 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
 ;
 ;READ UPDATE: ALL 0'S TO DATA STORE
 ;LOCATION 0000 FROM MAIN MEMORY
 ;LOC. 0
 MOV R0,(R1) ;WRITE MISS:NO UPDATE SHOULD OCCUR
 ;TO DATA STORE LOCATION 0000
 TST (R1) ;READ MISS;LOAD DATA STORE BITS RESULTING
 ;FROM PREVIOUS WRITE MISS INTO CDR<15:0>
 MOV CDR,R2 ;SAVE CDR CONTENTS
 MOV R0,(R1) ;WRITE HIT;
 ;THIS WRITE HIT SHOULD UPDATE DATA
 ;STORE LOCATION 0000.
 TST (R1) ;READ HIT;LOAD DATA STORE BITS RESULTING
 ;FROM PREVIOUS WRITE HIT INTO CDR<15:0>
 MOV CDR,R3 ;SAVE CDR CONTENTS
 ;
 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
 NOP ;FOR LOOP ON ERROR
 ;
 CLRB CMR ;DISABLE MAINTENANCE
 MOV #OFF,CCR ;DISABLE CACHE

024172 004437 002352
024176 024356

JSR R4,RELCTH
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

4510 024200 012737 177777 177752
4511 024206 112737 000002 177750
4512
4513
4514
4515 024214 012737 000015 177746
4516 024222 005737 040000
4517 024226 005737 000000
4518 024232 052737 000001 177750
4519
4520
4521 024240 005737 000000
4522
4523
4524 024244 005737 000000
4525
4526 024250 013737 177752 047310

1\$: MOV #-1,CHR
MOV B #HODO,CMR

;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
;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
BIS #TDAR,CMR

;READ UPDATE;LOAD TAG STORE WITH ALL 0'S
;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.
;SAVE CHR DATA

MOV CHR,CHR157

```
4528 024256 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
4529                                     25$:  NOP           ;INSTRUCTION 'JMP 1$' PLACED HERE
      024264 000240                                     NOP           ;FOR LOOP ON ERROR
      024266 000240

4530 024270 105037 177750      CLRB CMR          ;DISABLE MAINTENANCE MODE
4531 024274 042737 000177 047310      BIC #177,CHR157  ;INTERESTED IN 15:07
4532 024302 005737 047310      TST CHR157       ;BITS 15:07 SHOULD BE ALL 0'S
4533 024306 001422                                     BEQ 10$        ;PASS
4534 024310 012737 000007 002052      MOV #7,LOOP      ;ERROR;PREPARE CHR157 FOR TYPEOUT
4535 024316 006237 047310      2$:  ASR CHR157
4536 024322 042737 100000 047310      BIC #100000,CHR157
4537 024330 005337 002052      DEC LOOP
4538 024334 001370      BNE 2$
4539 024336 005037 047336      CLR EXDAT3      ;INDICATE EXPECTED DATA
4540                                     ERROR          ;ERROR
      024342 104000                                     ;-----
      024344 024342      .-2
4541                                     ;WRITE CONTROL LOGIC TESTS
4542                                     ;READING TAG STORE DATA THRU CHR<15:07>
4543                                     ;DID NOT RESULT IN ALL 0'S.
4544                                     ;THIS SUGGESTS THAT AN UPDATE OCCURED
4545                                     ;AND WRITE CONTROL LOGIC WAS NOT
4546                                     ;INHIBITED DUE TO READ HIT.
4547 024346 047336      EXDAT3          ;PRINT CHR<15:07> EXPECTED
4548 024350 047310      CHR157         ;PRINT CHR<15:07> RECEIVED
4549 024352 000000      0
4550 024354 000240      10$: NOP           ;END OF TEST
4551
4552
4553
4554
4555
4556 024356 005237 001464      *143: INC $TESTN ;UPDATE TEST ID
      ;*****
      .SBTTL
      .SBTTL TEST 143
      .SBTTL
4557 .SBTTL VERIFY THAT WRITE CONTROL LOGIC WILL INHIBIT A READ UPDATE
4558 .SBTTL TO CACHE TAG STORE DUE TO AN ACCESS TO I/O PAGE.
4559 ;*****
4560 024362 000004      SPCOND         ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      024364 024374      40$          ;ERROR/LOOP ON TEST
      024366 060000      1$-40$+60000-14 ;TEST START LOCATION
      024370 000000      0           ;LOOP ON ERROR START LOCATION
      024372 060042      25$-40$+60000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      024374 012737 001015 177746      40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
      024402 004437 002324      JSR R4,RELCTL  ;DISABLE CACHE
      024406 024556      10$+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

4561 024410 112737 000002 177750      1$:  MOVB #HODG,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
4562                                     ;WRITTEN TO CHR<15:07> ONLY DURING
```

```

4563                                     ;THE DESTINATION MEMORY ACCESS
4564                                     ;OF AN INSTRUCTION
4565 024416 012737 000015 177746      MOV #15,CCR
4566 024424 005737 057744            TST 57744
4567 024430 005737 077744            TST 77744
4568                                     ;READ UPDATE;LOAD BIT PATTERN
4569                                     ;000000011 INTO TAG STORE LOCATION
4570 024434 005737 177744            TST 17744
4571                                     ;7762
4572                                     ;ACCESS I/O PAGE BY READING CCR REGISTER.
4573                                     ;THE CACHE COULD DO AN UPDATE TO
4574                                     ;TAG STORE LOCATION 7762 BUT THE ACCESS
4575 024440 005737 057744            TST 57744
4576                                     ;TO I/O PAGE WILL INHIBIT WRITE CONTROL
4577 024444 013737 177752 047310     MOV CHR,CHR157
4578                                     ;LOGIC
4579 024452 000240                    25$: NOP
4580 024454 000240                    NOP
4581                                     ;INSTRUCTION 'JMP 1$' PLACED HERE
4582                                     ;FOR LOOP ON ERROR
4583 024456 105037 177750            CLRB CMR
4584 024462 012737 001015 177746     MOV #OFF,CCR
4585 024470 042737 000177 047310     BIC #177,CHR157
4586 024476 022737 000600 047310     CMP #600,CHR157
4587 024504 001423                    BEQ 10$
4588 024506 012737 000007 002052     MOV #7,LOOP
4589 024514 006237 047310            2$: ASR CHR157
4590 024520 042737 100000 047310     BIC #100000,CHR157
4591 024526 005337 002052            DEC LOOP
4592 024532 001370                    BNE 2$
4593 024534 012737 000003 047336     MOV #3,EXDAT3
4594 024542 104000                    ERROR
4595 024544 024542                    .-2
4596                                     ;INDICATE EXPECTED DATA
4597                                     ;ERROR
4598                                     ;-----
4599                                     ;WRITE CONTROL LOGIC TESTS
4600                                     ;READING TAGD<21:13> THRU CHR<15:07>
4601                                     ;DID NOT RESULT IN BIT PATTERN 000000011.
4602 024546 047336                    EXDAT3
4603 024550 047310                    CHR157
4604 024552 000000                    0
4605 024554 000240                    10$: NOP
4606                                     ;END OF TEST
    
```

```

4604 024556 005237 001464      T144: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 144
.SBTTL
.SBTTL THIS TEST VERIFIES THE AREA OF WRIT CONTROL LOGIC AND VALID
4605 .SBTTL STORE LOGIC WHICH IS CONCERNED WITH BYPASS OPERATIONS. A WIRE STRAP
4606 .SBTTL IS USED TO ALLOW OR INHIBIT INVALIDATION OF VALID STORE DURING
4607 .SBTTL READ BYPASS CONDITIONS. UNLESS SWITCH REGISTER 08 IS IMPLEMENTED,
4608 .SBTTL THIS TEST ASSUMES THAT STRAP W1 IS IN PLACE.
4609
    
```

```

4610 .SBTTL
4611 :
4612 :
4613 :
4614 :
4615 :
4616 :
4617 :
4618 :
4619 :
4620 :
4621 024562 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
024564 024574 40$ ;TEST START LOCATION
024566 070026 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
024570 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
024572 070112 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
024574 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
024602 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
024606 025030 10$+2 ;ADDRESS OF START OF NEXT TEST

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

4622 024610 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
4623 024616 001407 BEQ 1$ ;YES
4624 024620 052757 000400 177746 BIS #FC,CCR ;NO,FLUSH CACHE FOR SET A
4625 024626 032737 010000 177746 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
4626 024634 001374 BNE -6
4627 024636 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO ENABLE CACHE STORES
4628 024644 112737 000002 177750 MOVB #HODO,CMP ;HODO ALLOWS READ UPDATES,AND VALID
;STORE DATA TO BE WRITTEN TO CMR <12>
;ONLY DURING THE DESTINATION ACCESS
;OF AN INSTRUCTION.
4629
4630
4631
4632 024652 005737 040000 TST 40000
4633 024656 005737 060000 TST 60000 ;READ UPDATE TO CACHE LOCATION 0000.
4634
4635 ;WRITE 1 INTO VALID STORE LOCATION 0000.
4636 024662 052737 001000 177746 BIS #UCB,CCR ;BYPASS MODE
4637 024670 005737 040000 TST 40000 ;READ MISS/BYPASS;
4638 024674 005737 060000 TST 60000 ;LOAD VALID STORE LOCATION 0000 DATA
;RESULTING FROM PREVIOUS READ MISS/BYPASS
;INTO CMR<12>.
;THIS IS ALSO A READ HIT/BYPASS CONDITION.
4639
4640
4641
4642 024700 013700 177750 MOV CMR,R0 ;SAVE CMR CONTENTS
4643 024704 005737 040000 TST 40000 ;LOAD VALID STORE LOCATION 0000
;DATA RESULTING FROM PREVIOUS READ HIT
;/BYPASS INTO CMR<12>.
4644
4645
4646 024710 013701 177750 MOV CMR,R1 ;SAVE CMR CONTENTS
4647 024714 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4648
024722 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
024724 000240 NOP ;FOR LOOP ON ERROR

4649 024726 105037 177750 CLRB CMR ;DISABLE MAINTENANCE MODE
4650 024732 012702 002064 MOV #SWR,R2
4651 024736 032732 000400 BIT #BIT08,@(R2)+ ;IS STRAP W2 IN PLACE

```

```

4652
4653 024742 001015          BNE 7$          ;YES
4654 024744 032700 010000  BIT #VLD,R0    ;NO; W1 IS ASSUMED
4655 024750 001003          BNE 6$          ;PASS ;VALID DATA IS A 1
4656
    024752 104000          ERROR            ;ERROR
    ;-----
    024754 024752          .-2
4657
4658
4659
4660
4661 024756 000000          0
4662 024760 032701 010000  6$: BIT #VLD,R1  ;TEST VALID DATA HELD IN R1
4663 024764 001020          BNE 10$        ;PASS; STILL A 1
4664
    024766 104000          ERROR            ;ERROR
    ;-----
    024770 024766          .-2
4665
4666
4667
4668
4669 024772 000000          0
4670 024774 000414          BR 10$
4671 024776 032700 010000  7$: BIT #VLD,R0  ; W2 IS ASSUMED IN PLACE
4672 025002 001003          BNE 8$          ;PASS ;VALID DATA IS A 1
4673
    025004 104000          ERROR            ;ERROR
    ;-----
    025006 025004          .-2
4674
4675
4676
4677
4678 025010 000000          0
4679 025012 032701 010000  8$: BIT #VLD,R1  ;TEST VALID DATA HELD IN R1
4680 025016 001403          BEQ 10$        ;PASS; VALID DATA IS A 0
4681
    025020 104000          ERROR            ;ERROR
    ;-----
    025022 025020          .-2
4682
4683
4684
4685
4686 025024 000000          0
4687 025026 000240          10$: NOP        ;END OF TEST
4688
4689
4690
4691
4692
4693
4694 025030 005237 001464  T145:          INC $TESTN      ;UPDATE TEST ID
    ;*****
    ;SBTTL

```


.SBTTL TEST 145

.SBTTL

.SBTTL VERIFY THAT A WRITE HIT IN BYPASS MODE WILL INVALIDATE CACHE LOCATION

::*****

4695										
4696										
4697	025034	000004								
	025036	025046								
	025040	070026								
	025042	000000								
	025044	070102								
	025046	012737	001015	177746		40\$:	MOV #OFF,CCR			
	025054	004437	002352				JSR R4,RELCTH			
	025060	025212					10\$+2			

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

4698	025062	032737	020000	177746			BIT #VSIU,CCR			:IS SET A BEING USED?
4699	025070	001407					BEQ 1\$:YES
4700	025072	052737	000400	177746			BIS #FC,CCR			:NO,FLUSH CACHE FOR SET A
4701	025100	032737	010000	177746			BIT #VCIP,CCR			:WAIT TILL FLUSH COMPLETE
4702	025106	001374					BNE .-6			
4703	025110	012700	000002			1\$:	MOV #2,R0			:DATA TO R0
4704	025114	005001					CLR R1			:ADDRESS 0 TO R1
4705										
4706	025116	012737	000015	177746			MOV #15,CCR			:NO UCB SO AS TO ENABLE CACHE STORES
4707	025124	112737	000002	177750			MOVB #HODO,CMR			:HODO ALLOWS READ UPDATES,AND VALID
4708										:STORE DATA TO BE WRITTEN TO CMR <12>
4709										:ONLY DURING THE DESTINATION ACCESS
4710										:OF AN INSTRUCTION.
4711	025132	005737	040000				TST 40000			
4712	025136	005711					TST (R1)			:READ UPDATE TO CACHE LOCATION 0000.
4713										
4714										:WRITE 1 INTO VALID STORE LOCATION 0000.
4715	025140	052737	001000	177746			BIS #UCB,CCR			:BYPASS MODE
4716	025146	010011					MOV R0,(R1)			:WRITE HIT BYPASS TO LOC. 0;
4717										:SHOULD INVALIDATE
4718	025150	005711					TST (R1)			:LOAD VALID STORE LOCATION 0000 DATA
4719										:RESULTING FROM PREVIOUS WRITE HIT/BYPASS
4720										:INTO CMR<12>.
4721	025152	013702	177750				MOV CMR,R2			:SAVE CMR CONTENTS
4722	025156	012737	001015	177746			MOV #OFF,CCR			:DISABLE CACHE
4723										
	025164	000240				25\$:	NOP			:INSTRUCTION 'JMP 1\$' PLACED HERE
	025166	000240					NOP			:FOR LOOP ON ERROR
4724	025170	105037	177750				CLRB CMR			:DISABLE MAINTENANCE MODE
4725	025174	032702	010000				BIT #VLD,R2			:CHECK FOR 0
4726	025200	001403					BEQ 10\$:PASS
4727										
	025202	104000					ERROR			:ERROR
										:-----
	025204	025202					.-2			
4728										:WRITE CONTROL LOGIC TESTS
4729										:WRITE HIT /BYPASS DID NOT INVALIDATE
4730										:CACHE VALID STORE LOCATION
4731	025206	000000					0			

4732 025210 00G240 10\$: NOP ;END OF TEST

4733
4734
4735
4736
4737
4738
4739
4740
4741
4742

.SBTTL *
.SBTTL *
.SBTTL * CACHE DATA STORE RAM MEMORY MARCH PATTERN TEST
.SBTTL *
.SBTTL *

4743 025212 005237 001464

T146: INC \$TESTN ;UPDATE TEST ID

4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758

.SBTTL
.SBTTL TEST 146
.SBTTL
.SBTTL
.SBTTL VERIFY CACHE DATA STORE RAM MEMORY IC'S BY PERFORMING A
.SBTTL 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.

4759 025216 000004

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40\$;TEST START LOCATION
1\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25\$-40\$+70000-14 ;LOOP ON ERROR END LOCATION
40\$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
10\$+2 ;ADDRESS OF START OF NEXT TEST

025220 025230
025222 070000
025224 000000
025226 070100
025230 012737 001015 177746
025236 004437 002352
025242 025532

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

4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772

025244 012700 060000
025250 005020
025252 020027 070000
025256 001374
025260 012700 060000
025264 012701 040000
025270 012702 177777
025274 112737 000002 177750
025302 012737 000015 177746

1\$: MOV #60000,R0 ;ADDRESS LOC. 60000 TO R0
5\$: CLR (R0)+ ;CLEAR ALL LOW CACHE MAIN MEMORY
CMP R0,#70000 ;DONE?
BNE 5\$;NO
MOV #60000,R0 ;ADDR. LOC. 60000 TO R0
MOV #40000,R1 ;ADDR. LOC. 40000 TO R1
MOV #-1,R2 ;R2 CONTAINS ALL 1'S
MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & DATA STORE BITS
;TO BE WRITTEN TO CDR<15:0> ONLY DURING
;THE DESTINATION MEMORY ACCESS OF AN
;INSTRUCTION
MOV #15,CCR ;WRITE ENABLE CACHE DATA STORES

4913
4914
4915
4916
4917
4918 025732 047320
4919 025734 047316
4920
4921 025736 047322
4922 025740 000000
4923 025742 000435
4924 025744 022703 177777
4925 025750 001425
4926 025752 012737 001015 177746
4927 025760 105037 177750
4928 025764 012737 177777 047320
4929 025772 010337 047316
4930
4931 025776 010037 047322
4932 026002 006237 047322
4933
026006 104000
026010 026006
4934

EXDAT6
CDR150
CA121
0
BR 3\$
8\$: CMP #-1,R3
BEQ 9\$
MOV #OFF,CCR
CLRB CMR
MOV #-1,EXDAT6
MOV R3,CDR150
MOV R0,CA121
ASR CA121
ERROR
.-?

:WITH A 1 WHEN WRITING A 1 TO ANOTHER
:LOCATION.ANY BIT IN CDR150 DATA
:THAT IS A 1 MAY POINT TO A BAD
:CACHE DATA STORE RAM.
: EXPECTED CACHE DATA STORE DATA
: CACHE DATA STORE DATA READ
:THRU CDR<15:0>
:SPECIFY FAILED DATA STORE ADDRESS LOCATION
:END THE TEST
:SHOULD BE ALL 1'S
:PASS
:DISABLE CACHE
:CLEAR MAINT. MODE
:SPECIFY EXPECTED CACHE DATA STORE DATA
:SPECIFY CACHE DATA STORE DATA READ
:THRU CDR<15:0>
:SPECIFY FAILED DATA STORE ADDRESS LOCATION
:ERROR
:-----
:DATA STORE MARCH PATTERN TEST

```

4936 ;READING CACHE DATA STORE DATA
4937 ;THRU CDR<15:0> DID NOT READ ALL 1'S.
4938 ;ANY BIT IN CDR150 DATA
4939 ;THAT IS A 0 MAY POINT TO A BAD
4940 ;CACHE DATA STORE RAM.
4941
4942 026012 047320 EXDAT6 ; EXPECTED CACHE DATA STORE DATA
4943 026014 047316 CDR150 ; CACHE DATA STORE DATA READ
4944 ;THRU CDR<15:0>
4945 026016 047322 CA121 ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4946 026020 000000 0
4947 026022 000405 BR 3$ ;END TEST
4948 026024 062700 000002 9$: ADD #2,RO ;NEXT LOCATION
4949 026030 022700 100000 CMP #100000,RO ;HAS ALL HI CACHE BEEN DONE?
4950 026034 001304 BNE 7$ ;NO,CONTINUE
4951 026036 012737 001015 177746 3$: MOV #OFF,CCR ;DISABLE CACHE
4952 026044 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
4953 026050 000240 10$: NOP ;END OF TEST
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965 026052 005237 001464

```

```

.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *
.SBTTL *

```

CACHE TAG STORE RAM MEMORY MARCH PATTERN TEST

```

T150: INC $TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 150
.SBTTL
.SBTTL
.SBTTL VERIFY CACHE TAG STORE RAM MEMORY IC'S BY PERFORMING A
.SBTTL MARCH PATTERN TEST TO LOW CACHE AREA OF TAG STORE(LOC. 0000-3777)
:*****
PROCEDURE: 1. WRITE ALL 0'S TO ALL LO CACHE TAG STORE
RAMS CORRESPONDING TO LOCATIONS 0000-3777
2. READ 0'S FROM ALL LO CACHE RAMS CORRESPONDING
TO LOCATION 0000
3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
0000.
4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
0000.
5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
AND UNTIL LOC. 3777 IS REACHED.
:*****
4981 026056 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
026060 026070 40$ ;TEST START LOCATION
026062 070000 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
026064 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
026066 070070 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
026070 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
026076 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE

```

```

026102 026446          10$+2          ;ADDRESS OF START OF NEXT TEST

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

4982
4983 026104 005037 177752      1$: CLR CHR          ;LOAD AMR<8:0> BY WRITING ALL 0'S TO
4984                                ;CHR<8:0>
4985 026110 012700 060000      MOV #60000,R0        ;ADDR. LOC. 60000 TO R0
4986 026114 012701 040000      MOV #40000,R1        ;ADDR. LOC. 40000 TO R1
4987 026120 112737 000003 177750  MOVB #HODO+TDAR,CMR ;HCDO ALLOWS CACHE UPDATES & TAG STORE BITS
4988                                ;TO BE WRITTEN TO CHR<15:7> ONLY DURING
4989                                ;THE DESTINATION MEMORY ACCESS OF AN
4990                                ;INSTRUCTION
4991                                ;TDAR ALLOWS TAG FIELD TO BE WRITTEN
4992                                ;INTO FROM AMR<8:0>
4993 026126 012737 000015 177746  MOV #15,CCR          ;WRITE ENABLE CACHE TAG STORES
4994
4995 026134 005721      6$: *ST (R1)+          ;WRITE ALL LOW CACHE
4996                                ;TAG STORE WITH 0'S
4997 026136 005720      TST (R0)+          ;
4998 026140 020027 070000      CMP R0,#70000        ;DONE?
4999 026144 001373      BNE 6$              ;NO
5000
5001 026146 012737 177777 177752  MOV #-1,CHR          ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
5002 026154 012700 060000      MOV #60000,R0        ;ADDR. 60000 TO R0
5003 026160 005710      7$: TST (R0)          ;READ MISS TO CACHE TAG STORE LOCATION
5004                                ;SPECIFIED BY R0.CLOCK TAG STORE
5005                                ;BITS INTO CHR<15:7>.SHOULD BE ALL 0'S.
5006                                ;ALSO CAUSES TAG STORE LOCATION TO BE
5007                                ;WRITTEN WITH 1'S FROM AMR<8:0>
5008 026162 013705 177752      MOV CHR,R5          ;SAVE CHR CONTENTS
5009 026166 005710      TST (R0)          ;READ MISS.CLOCK TAG STORE BITS TO
5010                                ;CHR <15:0>.SHOULD BE ALL 1'S.
5011 026170 013703 177752      MOV CHR,R3          ;SAVE CHR CONTENTS
5012
026174 000240      25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
026176 000240      NOP          ;FOR LOOP ON ERROR

5013 026200 042705 000177      BIC #177,R5
5014 026204 005705      TST R5              ;SHOULD BE ALL 0'S
5015 026206 001437      BEQ 8$              ;PASS
5016 026210 012737 001015 177746  MOV #OFF,CCR        ;DISABLE CACHE
5017 026216 105037 177750      CLRB CMR           ;CLEAR MAINT. MODE
5018 026222 005037 047336      CLR EXDAT3         ;SPECIFY EXPECTED CACHE TAG STORE DATA
5019 026226 010537 047310      MOV R5,CHR157      ;SPECIFY CACHE TAG STORE TAG READ
5020                                ;THRU CHR<15:7>
5021 026232 012737 000007 002052  MOV #7,LOOP        ;PREPARE CHR157 FOR TYPEOUT
5022 026240 006237 047310      4$: ASR CHR157
5023 026244 042737 100000 047310  BIC #100000,CHR157
5024 026252 005337 002052      DEC LOOP
5025 026256 001370      BNE 4$
5026
5027 026260 010037 047322      MOV R0,CA121        ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
5028 026264 006237 047322      ASR CA121
5029
026270 104000      ERROR              ;ERROR

```



```

026272 026270      .-2
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040 026274 047336      EXDAT3
5041 026276 047310      CHR157
5042
5043 026300 047322      CA121
5044 026302 000000      0
5045 026304 000452      BR 3$
5046 026306 042703 000177      8$: BIC #177,R3
5047
5048 026312 022703 177600      CMP #177600,R3
5049 026316 001440      BEQ 9$
5050 026320 012737 001015 177746      MOV #OFF,CCR
5051 026326 105037 177750      CLRB CMR
5052 026332 012737 177777 047336      MOV #-1,EXDAT3
5053 026340 010337 047310      MOV R3,CHR157
5054
5055 026344 012737 000007 002052      MOV #7,LOOP
5056 026352 006237 047310      5$: ASR CHR157
5057 026356 042737 100000 047310      BIC #100000,CHR157
5058 026364 005337 002052      DEC LOOP
5059 026370 001370      BNE 5$
5060
5061 026372 010037 047322      MOV R0,CA121
5062 026376 006237 047322      ASR CA121
5063
026402 104000      ERROR
026404 026402      .-2
5064
5065
5066
5067
5068
5069
5070
5071 026406 047336      EXDAT3
5072
5073
5074 026410 047310      CHR157
5075 026412 047322      CA121
5076 026414 000000      0
5077 026416 000405      BR 3$
5078 026420 062700 000002      9$: ADD #2,R0
5079 026424 022700 070000      CMP #70000,R0
5080 026430 001253      BNE 7$
5081 026432 012737 001015 177746      3$: MOV #OFF,CCR

```

```

;-----
;TAG STORE MARCH PATTERN TEST
;READING CACHE TAG STORE DATA
;THRU CHR<15:7> 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 CHR157 DATA
;THAT IS A 1 MAY POINT TO A BAD
;CACHE TAG STORE RAM.

```

```

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

```

```

;END THE TEST
;PREPARE R3 FOR CHECK

```

```

;SHOULD BE ALL 1'S
;PASS
;DISABLE CACHE
;CLEAR MAINT. MODE
;SPECIFY EXPECTED CACHE TAG STORE DATA
;SPECIFY CACHE TAG STORE DATA READ
;THRU CHR<15:7>
;PREPARE CHR157 FOR TYPEOUT

```

```

;SPECIFY FAILED TAG STORE ADDRESS LOCATION

```

```

;ERROR
;-----

```

```

;TAG STORE MARCH PATTERN TEST
;READING CACHE TAG STORE DATA
;THRU CHR<15:7> DID NOT READ ALL 1'S.
;ANY BIT IN CHR157 DATA
;THAT IS A 0 MAY POINT TO A BAD
;CACHE TAG STORE RAM.

```

```

; EXPECTED CACHE TAG STORE DATA
; 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

```

5082 026440 105037 177750
5083 026444 000240
5084
5085
5086
5087
5088
5089
5090 026446 005237 001464

10\$: CLR B CMR ;DISABLE MAINT. MODE
NOP ;END OF TEST

T151: INC \$TESTN ;UPDATE TEST ID
:*****

5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106 026452 000004

.SBTTL
.SBTTL TEST 151
.SBTTL
.SBTTL
.SBTTL VERIFY CACHE TAG STORE RAM MEMORY IC'S BY PERFORMING A
.SBTTL 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.

:*****
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$;ERROR/LOOP ON TEST
1\$-40\$+60000-14 ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25\$-40\$+60000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTL ;DISABLE CACHE
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

5107
5108 026500 005037 177752
5109
5110 026504 012700 070000
5111 026510 012701 050000
5112 026514 112737 000003 177750
5113
5114
5115
5116
5117
5118 026522 012737 000015 177746
5119
5120 026530 005721
5121
5122 026532 005720

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
MOV B #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>
MOV #15,CCR ;WRITE ENABLE CACHE TAG STORES
6\$: TST (R1)+ ;WRITE ALL HIGH CACHE
;TAG STORE WITH 0'S
TST (R0)+ ;

```

5123 026534 020027 100000          CMP R0,#100000          ;DONE?
5124 026540 001373                BNE 6$                 ;NO
5125
5126 026542 012737 177777 177752  MOV #-1,CHR           ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
5127 026550 012700 070000          MOV #70000,R0         ;ADDR. 70000 TO R0
5128 026554 005710                7$: TST (R0)            ;READ MISS TO CACHE TAG STORE LOCATION
5129
5130
5131
5132
5133 026556 013705 177752          MOV CHR,R5            ;SPECIFIED BY R0.CLOCK TAG STORE
5134 026562 005710                TST (R0)              ;BITS INTO CHR<15:7>.SHOULD BE ALL 0'S.
5135
5136 026564 013703 177752          MOV CHR,R3            ;ALSO CAUSES TAG STORE LOCATION TO BE
5137
5137 026570 000240                25$: NOP               ;WRITTEN WITH 1'S FROM AMR<8:0>
5137 026572 000240                NOP                    ;SAVE CHR CONTENTS
;READ MISS.CLOCK TAG STORE BITS TO
;CHR <15:7>.SHOULD BE 'ALL 1'S.
;SAVE CHR CONTENTS
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR

5138 026574 042705 000177          BIC #177,R5
5139 026600 005705                TST R5                ;SHOULD BE ALL 0'S
5140 026602 001437                BEQ 8$                 ;PASS
5141 026604 012737 001015 177746  MOV #OFF,CCR          ;DISABLE CACHE
5142 026612 105037 177750          CLR B CMR             ;CLEAR MAINT. MODE
5143 026616 005037 047336          CLR EXDAT3            ;SPECIFY EXPECTED CACHE TAG STORE DATA
5144 026622 010537 0473'0         MOV R5,CHR157         ;SPECIFY CACHE TAG STORE TAG READ
5145
5146 026626 012737 000007 002052  MOV #7,LOOP           ;THRU CHR<15:7>
5147 026634 006237 047310                4$: ASR CHR157         ;PREPARE CHR157 FOR TYPEOUT
5148 026640 042737 100000 047310  BIC #100000,CHR157
5149 026646 005337 002052          DEC LOOP
5150 026652 001370                BNE 4$
5151
5152 026654 010037 047322          MOV R0,CA'21          ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
5153 026660 006237 047322          ASR CA121
5154
5154 026664 104000                ERROR                  ;ERROR
;-----
5154 026666 026664                .-2
;TAG STORE MARCH PATTERN TEST
;READING CACHE TAG STORE DATA
;THRU CHR<15:7> 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 CHR157 DATA
;THAT IS A 1 MAY POINT TO A BAD
;CACHE TAG STORE RAM.

5165 026670 047336          EXDAT3                ; EXPECTED CACHE TAG STORE DA*A
5166 026672 047310          CHR157                ; CACHE TAG STORE DATA READ
5167
5168 026674 047322          CA121                 ;THRU CHR<15:7>
5169 026676 000000          0                     ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
5170 026700 000452
5171 026702 042703 000177                8$: BR 3$
5172
5173 026706 022703 177600          BIC #177,R3           ;END THE TEST
;PREPARE R3 FOR CHECK
;SHOULD BE ALL 1'S

```

```
5174 026712 001440 BEQ 9$ ;PASS
5175 026714 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
5176 026722 105037 177750 CLR B CMR ;CLEAR MAINT. MODE
5177 026726 012737 177777 047336 MOV #-1,EXDAT3 ;SPECIFY EXPECTED CACHE TAG STORE DATA
5178 026734 010337 047310 MOV R3,CHR157 ;SPECIFY CACHE TAG STORE DATA READ
5179 ;THRU CHR<15:7>
5180 026740 012737 000007 002052 MOV #7,LOOP ;PREPARE CHR157 FOR TYPEOUT
5181 026746 006237 047310 5$: ASR CHR157
5182 026752 042737 100000 047310 BIC #100000,CHR157
5183 026760 005337 002052 DEC LOOP
5184 026764 001370 BNE 5$
5185
5186 026766 010037 047322 MOV R0,CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
5187 026772 006237 047322 ASR CA121
5188
5188 026776 104000 ERROR ;ERROR
5188 027000 026776 .-2 ;-----
5189 ;TAG STORE MARCH PATTERN TEST
5190 ;READING CACHE TAG STORE DATA
5191 ;THRU CHR<15:7> DID NOT READ ALL 15'S.
5192 ;ANY BIT IN CHR157 DATA
5193 ;THAT IS A 0 MAY POINT TO A BAD
5194 ;CACHE TAG STORE RAM.
5195
5196 027002 047336 EXDAT3 ; EXPECTED CACHE TAG STORE DATA
5197 ; CACHE TAG STORE DATA READ
5198 ;THRU CHR<15:7>
5199 027004 047310 CHR157 ;CACHE TAG STORE READ THRU CHR<15:7>
5200 027006 047322 CA121 ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
5201 027010 000000 0
5202 027012 000405 BR 3$ ;END TEST
5203 027014 062700 000002 9$: ADD #2,R0 ;NEXT LOCATION
5204 027020 022700 100000 CMP #100000,R0 ;HAS ALL HI CACHE BEEN DONE?
5205 027024 001253 BNE 7$ ;NO,CONTINUE
5206 027026 012737 001015 177746 3$: MOV #OFF,CCR ;DISABLE CACHE
5207 027034 105037 177750 CLR B CMR ;DISABLE MAINT. MODE
5208 027040 000240 10$: NOP ;END OF TEST
5209
5210
5211
5212
5213
5214
5215 .SBTTL *
5216 .SBTTL
5217 .SBTTL *
5218 .SBTTL
5219 .SBTTL
5220 .SBTTL *
5221
5222
5223
5224
5225 027042 005237 001464 T152: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
```

.SBTTL TEST 152

.SBTTL

.SBTTL VERIFY THAT LOW AND HI BYTE DATA PARITY STORES CAN HOLD A 0 AT DATA
.SBTTL PARITY STORE LOCATION 0000.

.SBTTL

PROCEDURE: GENERATE 0'S FROM UPPER AND LOWER BYTE PARITY
DATA GENERATORS BY PLACING ALL C 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:
WRD<15:0> ALL 0'S
WVPD(1)= 0
DATA PARITY STORE ADDRESS:
CA<12:1>=0000

RESULT: CMR<11:10> BOTH 0

5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243

5244 027046 000004

027050 027060
027052 070004
027054 000000
027056 070040
027060 012737 001015 177746
027066 004437 002352
027072 027204

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40\$;TEST START LOCATION
1\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25\$-40\$+70000-14 ;LOOP ON ERROR END LOCATION
40\$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
10\$+2 ;ADDRESS OF START OF NEXT TEST

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

5245 027074 005037 060000
5246 027100 1:2737 000002 177750
5247
5248
5249
5250 027106 012737 000015 177746
5251
5252 027114 005737 040000
5253 027120 005737 060000
5254
5255 027124 005737 060000
5256
5257 027130 013701 177750
5258
027134 000240
027136 000240
5259 027140 105037 177750
5260 027144 012737 001015 177746
5261 027152 032701 004000
5262 027156 001403
5263
027160 104000

1\$: CLR 60000 ;0'S TO MAIN MEMORY LOCATION
MOV B #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 ALL 0'S ON WRD<15:0> INPUTS
;THEREBY WRITING 0 INTO PARITY STORE LOCATION 0000.
TST 60000 ;WRITE UPPER AND LOWER DATA PARITY BITS FROM
;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
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 #HPB,R1 ;CHECK FOR 0
BEQ 9\$;PASS
ERROR ;ERROR
;-----

```

5264 027162 027160      .-2      ;DATA PARITY GEN. & STORE TESTS
5265                      ;READING CACHE MAINT. REGISTER
5266                      ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5267                      ;NOT RESULT IN 0.
5268 027164 000000      0
5269      :::
5270 027166 032701 002000 9$: BIT #LPB,R1      ;CHECK 0 FOR LOWER BYTE PARITY DATA
5271 027172 001403      BEQ 10$      ;PASS
5272                      ;
5273 027174 104000      ERROR      ;ERROR
5274                      ;-----
5275 027176 027174      .-2
5276                      ;DAT. PARITY GEN. & STORE TESTS
5277                      ;READING CACHE MAINT. REGISTER
5278                      ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5279                      ;NOT RESULT IN 0.
5280
5281 027200 000000      0
5282 027202 000240      10$: NOP      ;END OF TEST
5283
5284 027204 005237 001464 T153: INC $TESTN      ;UPDATE TEST ID
5285      ;*****
5286      .SBTTL
5287      .SBTTL TEST 153
5288      .SBTTL
5289      .SBTTL VERIFY THAT THE LOW BYTE DATA PARITY GENERATOR WILL WRITE A 1 INTO ADDRESS LOCATION
5290      .SBTTL FOR FLOATING 1 ACROSS 0 DATA PATTERN ON DATA PARITY GENERATOR INPUTS.
5291      .SBTTL
5292      ;
5293      ;PROCEDURE: FOR EACH FLOATING 1 PATTERN READ DATA PARITY STORE BITS
5294      ;FROM CMR<11:10>
5295      ;
5296      ;CONDITIONS:
5297      ;INPUTS TO DATA PARITY GEN.:
5298      ;WRTD<7:0> FLOATING 1 ACROSS 0'S
5299      ;WRTD<15:8> ALL 0'S
5300      ;WWPD(1)=0
5301      ;DATA PARITY STORE ADDRESS:
5302      ;CA<12:1>-0000
5303      ;
5304      ;RESULT: CMR<11>=0
5305      ;CMR<10>=1
5306      ;*****
5307 027210 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
5308                      ;ERROR/LOOP ON TEST
5309 027212 027222      40$      ;TEST START LOCATION
5310 027214 070014      1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
5311 027216 000000      0      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
5312 027220 070050      25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
5313 027222 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
5314 027230 004437 002352      JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
5315 027234 027376      10$+2      ;ADDRESS OF START OF NEXT TEST

```

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

```

5301 027236 012737 000001 047332          MOV #1,FLTPAT          ;1ST FLOATING 1 PATTERN:000001
5302 027244 013737 047332 060000          2$: MOV FLTPAT,60000    ;FLOATING PATTERN TO MAIN MEMORY
5303 027252 112737 000002 177750          1$: MOVB #HODO,CMR     ;ALLOWS UPPER AND LOWER BYTE DATA
5304                                     ;PARITY STORE BITS TO BE WRITTEN TO
5305                                     ;CMR<11:10> ONLY DURING THE DESTINATION
5306                                     ;ACCESS OF AN INSTRUCTION.
5307 027260 012737 000015 177746          MOV #15,CCR           ;NO UCB SO AS TO WRITE ENABLE PARITY
5308                                     ;STORE
5309 027266 005737 040000          TST 40000
5310 027272 005737 060000          TST 60000
5311                                     ;PLACE FLOATING 1 PATTERN ON WRTD<15:0> INPUTS
5312                                     ;THEREBY WRITING 1 IN LOW BYTE AND 0 IN HI
5313 027276 005737 060000          TST 60000            ;BYTE DATA PARITY STORE LOCATION 0000.
5314                                     ;WRITE UPPER AND LOWER BYTE DATA PARITY BITS FROM
5315 027302 013701 177750          MOV CMR,R1           ;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
5316                                     ;SAVE CMR DATA
          027306 000240          25$: NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
          027310 000240          NOP                  ;FOR LOOP ON ERROR

5317 027312 105037 177750          CLRB CMR             ;DISABLE MAINT. MODE
5318 027316 012737 001015 177746          MOV #OFF,CCR         ;DISABLE CACHE
5319 027324 032701 004000          BIT #HPB,R1         ;CHECK 0 FOR UPPER BYTE PARITY STORE
5320 027330 001404          BEQ 8$              ;PASS
5321          027332 104000          ERROR                ;ERROR
          027334 027332          .-2                  ;-----

5322                                     ;DATA PARITY GEN. & STORE TESTS
5323                                     ;READING CACHE MAINT. REGISTER
5324                                     ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5325                                     ;NOT RESULT IN 0.
5326 027336 047332          FLTPAT              ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
5327                                     ;GENERATOR INPUTS: WRTD<15:0>
5328 027340 000000          0
5329                                     ;:
5330 027342 032701 002000          8$: BIT #LPB,R1      ;CHECK 1 FOR LOWER BYTE PARITY DATA
5331 027346 001004          BNE 9$              ;PASS
5332          027350 104000          ERROR                ;ERROR
          027352 027350          .-2                  ;-----

5333                                     ;DAT. PARITY GEN. $ STORE TESTS
5334                                     ;READING CACHE MAINT. REGISTER
5335                                     ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5336                                     ;NOT RESULT IN 1.
5337 027354 047332          FLTPAT              ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
5338                                     ;GEN. INPUT: WRTD<15:0>
5339 027356 000000          0
5340                                     ;:
5341 027360 006337 047332          9$: ASL FLTPAT       ;NEXT PATTREN
5342 027364 032737 000400 047332          BIT #400,FLTPAT     ;HAS PATTERN 000200 BEEN DONE
5343 027372 001724          BEQ 2$              ;NO
5344 027374 000240          10$: NOP            ;END OF TEST
5345
5346
5347 027376 005237 001464          T154: INC $TESTN    ;UPDATE TEST ID
          ;*****

```

```

.SBTTL
.SBTTL TEST 154
.SBTTL
.SBTTL VERIFY THAT THE HI BYTE DATA PARITY GENERATOR WILL WRITE A 1 INTO ADDRESS LOCATION
.SBTTL FOR FLOATING 1 ACROSS 0 DATA PATTERN ON DATA PARITY GENERATOR INPUTS.
.SBTTL
          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
          WWPD(1)=0
          DATA PARITY STORE ADDRESS:
          CA<12:1>=0000
          RESULT: CMR<11>=1
                  CMR<10>=0
          *****
5348          027402  000004          SCPLND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
5349          027404  027414          40$            ;ERROR/LOOP ON TEST
5350          027406  070014          1$-40$+70000-14 ;TEST START LOCATION
5351          027410  000000          0              ;LOOP ON ERROR START LOCATION
5352          027412  070050          25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
5353          027414  012737  001015  177746  40$:          MOV #OFF,CCR    ;LOOP ON ERROR END LOCATION
5354          027422  004437  002352          JSR R4,RELCTH   ;DISABLE CACHE
5355          027426  027562          10$+2          ;LOCATE TEST CODE TO HIGH CACHE SPACE
5356          ;THE FOLLOWING LOCATIONS INCLUDING 10$
5357          ;ARE RELOCATED TO HI CACHE SPACE
5358
5359          5367 027430  012737  000400  047332          MOV #400,FLTPAT ;1ST FLOATING 1 PATTERN:000400
5360          5368 027436  013737  047332  060000          MOV FLTPAT,60000 ;FLOATING PATTERN TO MAIN MEMORY
5361          5369 027444  112737  000002  177750          1$:          MOVB #HODO,CMR   ;ALLOWS UPPER AND LOWER BYTE DATA
5362          5370          ;PARITY STORE BITS TO BE WRITTEN TO
5363          5371          ;CMR<11:10> ONLY DURING THE DESTINATION
5364          5372          ;ACCESS OF AN INSTRUCTION.
5365          5373 027452  012737  000015  177746          MOV #15,CCR     ;NO UCB SO AS TO WRITE ENABLE PARITY
5366          5374          ;STORE
5367          5375 027460  005737  040000          TST 40000
5368          5376 027464  005737  060000          TST 60000
5369          5377          ;PLACE FLOATING 1 PATTERN ON WRTD<15:0> INPUTS
5370          5378          ;THEREBY WRITING 0 IN LOW BYTE AND 1 IN HI
5371          5379 027470  005737  060000          TST 60000
5372          5380          ;BYTE DATA PARITY STORE LOCATION 0000.
5373          5381 027474  013701  177750          MOV CMR,R1     ;WRITE UPPER AND LOWER BYTE DATA PARITY BITS FROM
5374          5382          ;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
5375          027500  000240          25$:          NOP           ;SAVE CMR DATA
5376          027502  000240          NOP           ;INSTRUCTION 'JMP 1$' PLACED HERE
5377          5383 027504  105037  177750          CLRB CMR
5378          5384 027510  012737  001015  177746          MOV #OFF,CCR
5379          5385 027516  032701  004000          BIT #1,PB,R1
5380          5386 027522  001004          BNE 8$
          ;DISABLE MAINT. MODE
          ;DISABLE CACHE
          ;CHECK 1 FOR UPPER BYTE PARITY STORE
          ;PASS

```



```

5387
5388 027524 104000 ERROR ;ERROR
;-----
027526 027524 .-2
5389 ;DATA PARITY GEN. & STORE TESTS
5390 ;READING CACHE MAINT. REGISTER
5391 ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5392 ;NOT RESULT IN 1.
5393 027530 047332 FLTPAT ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
5394 ;GENERATOR INPUTS: WRD<15:0>
5395 027532 000000 0
5396 ;;;
5397 027534 032701 002000 8$: BIT #LPB,R1 ;CHECK 0 FOR LOWER BYTE PARITY DATA
5398 027540 001404 BEQ 9$ ;PASS
5399
5400 027542 104000 ERROR ;ERROR
;-----
027544 027542 .-2
5401 ;DAT. PARITY GEN. $ STORE TESTS
5402 ;READING CACHE MAINT. REGISTER
5403 ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5404 ;NOT RESULT IN 0.
5405 027546 047332 FLTPAT ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
5406 ;GEN. INPUTS: WRD<15:0>
5407 027550 000000 0
5408 ;;;
5409 027552 006337 047332 9$: ASL FLTPAT ;NEXT PATTERN
5410 027556 103327 BCC 2$ ;CONTINUE IF PATTERN 100000 NOT DONE
5411 027560 000240 10$: NOP ;END OF TEST
5412
5413
5414 027562 005237 001464 T155: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 155
.SBTTL
5415 .SBTTL VERIFY WRITE WRONG PARITY TO UPPER AND LOWER BYTE DATA PARITY
5416 .SBTTL STORE .
5417 .SBTTL
5418 ;;;
5419 ;;; CONDITIONS:
5420 ;;; INPUTS TO DATA PARITY GEN:
5421 ;;; WRD<15:0> ALL 0'S
5422 ;;; WWPD(1)= 1
5423 ;;; DATA PARITY STORE ADDRESS:
5424 ;;; CA<12:1>=0000
5425 ;;;
5426 ;;; RESULT: (MR<11:10> BOTH 1
5427 ;;;*****
5428 027566 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
027570 027600 40$ ;ERROR/LOOP ON TEST
027572 070004 1$-40$+70000-14 ;TEST START LOCATION
027574 000000 0 ;LOOP ON ERROR START LOCATION
027576 070040 25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION

```

```

027600 012737 001015 177746      40$:  MOV #OFF,CCR      ;DISABLE CACHE
027606 004437 002352              JSR R4,RELCTH      ;LOCATE TEST CODE TO HIGH CACHE SPACE
027612 027742 10$+2              ;ADDRESS OF START OF NEXT TEST

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

5429 027614 005037 060000      1$:  CLR 60000          ;0'S TO MAIN MEMORY LOCATION
5430 027620 112737 000002 177750  MOVB #HODO,CMR      ;ALLOWS UPPER AND LOWER BYTE DATA
5431                                     ;PARITY STORE BITS TO BE WRITTEN TO
5432                                     ;CMR<11:10> ONLY DURING THE DESTINATION
5433                                     ;ACCESS OF AN INSTRUCTION.
5434 027626 012737 000115 177746  MOV #15+WVPD,CCR    ;NO UCB SO AS TO WRITE ENABLE PARITY.
5435                                     ;ENABLE WRITE WRONG PARITY DATA
5436 027634 005737 040000      TST 40000           ;
5437 027640 005737 060000      TST 60000           ;PLACE ALL 0'S ON WRTD<15:0> INPUTS
5438                                     ;SINCE WVPD IS INVOKED A 1 WILL BE
5439                                     ;WRITTEN INTO PARITY STORE LOCATION 0000.
5440 027644 005737 060000      TST 60000           ;WRITE UPPER AND LOWER DATA PARITY BITS FROM
5441                                     ;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
5442 027650 013701 177750      MOV CMR,R1         ;SAVE CMR DATA
5443
027654 000240      25$:  NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
027656 000240      NOP              ;FOR LOOP ON ERROR

5444 027660 012737 001015 177746  MOV #OFF,CCR      ;DISABLE CACHE
5445 027666 105037 177750      CLRB CMR          ;DISABLE MAINT. MODE
5446 027672 052737 000400 177746  BIS #FC,CCR       ;BEFORE LEAVING TEST FLUSH CACHE TO
5447                                     ;REMOVE ANY EFFECTS OF WVPD
5448 027700 032737 010000 177746  BIT #VCIP,CCR     ;WAIT TILL DONE
5449 027706 001374      BNE .-6
5450 027710 032701 004000      BIT #HPB,R1       ;CHECK 1 FOR UPPER BYTE PARITY STORE.
5451 027714 001003      BNE 9$            ;PASS
5452
5453
027716 104000      ERROR              ;ERROR
;-----
027720 027716      .-2
;DATA PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 11 FOR UPPER BYTE PARITY DATA DID
;NOT RESULT IN 1.
5454
5455
5456
5457
5458 027722 000000      0
5459                                     ;
5460 027724 032701 002000      :::  9$:  BIT #LPB,R1       ;CHECK 1 FOR LOWER BYTE PARITY DATA
5461 027730 001003      BNE 10$          ;PASS
5462
5463
027732 104000      ERROR              ;ERROR
;-----
027734 027732      .-2
;DAT. PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 10 FOR LOWER BYTE PARITY DATA DID
;NOT RESULT IN 1.
5464
5465
5466
5467
5468 027736 000000      0
5469 027740 000240      10$:  NOP              ;END OF TEST
    
```

5470
 5471
 5472
 5473
 5474
 5475
 5476
 5477
 5478 027742 005237 001464

T156: INC \$TESTN ;UPDATE TEST ID
 :*****

.SBTTL
 .SBTTL TEST 156
 .SBTTL
 .SBTTL VERIFY THAT TAG PARITY STORE CAN HOLD A 0 AT TAG
 .SBTTL PARITY STORE LOCATION 0000.
 .SBTTL

5479
 5480
 5481
 5482
 5483
 5484
 5485
 5486
 5487
 5488
 5489
 5490
 5491
 5492
 5493
 5494
 5495
 5496

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> AL 0'S
 WWPT(1)= 0
 TAG PARITY STORE ADDRESS:
 CA<12:1>-0000

 RESULT: CMR<9>- 0

5497 027746 000004

 027750 027760
 027752 070004
 027754 000000
 027756 070040
 027760 012737 001015 177746
 027766 004437 002352
 027772 030070

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 ;ERROR/LOOP ON TEST
 40\$;TEST START LOCATION
 1\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
 25\$-40\$+70000-14 ;LOOP ON ERROR END LOCATION
 40\$: MOV #OFF,CCR ;DISABLE CACHE
 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

5498 027774 005037 177752
 5499 030000 112737 000003 177750
 5500
 5501
 5502
 5503
 5504
 5505 030006 012737 000015 177746
 5506
 5507 030014 005737 040000
 5508 030020 005737 060000
 5509
 5510 030024 005737 060000

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
 ;
 TST 40000
 TST 60000 ;PLACE ALL 0'S ON TAG WRTD<21:13> INPUTS
 ;THEREBY WRITING 0 INTO PARITY STORE LOCATION 0000.
 TST 60000 ;WRITE TAG PARITY BITS FROM

```

5511                                     ;LOCAT. 0000 INTO CMR<9>
5512 030030 013701 177750             MOV CMR,R1          ;SAVE CMR DATA
5513                                     ;INSTRUCTION 'JMP 1$' PLACED HERE
                                25$:  NOP                    ;FOR LOOP ON ERROR
                                030034 000240
                                030036 000240

5514 030040 105037 177750             CLRB CMR           ;DISABLE MAINT. MODE
5515 030044 012737 001015 177746     MOV #OFF,CCR      ;DISABLE CACHE
5516 030052 032701 001000             BIT #TPB,R1      ;CHECK FOR 0
5517 030056 001403                     BEQ 10$          ;PASS
5518
5519 030060 104000                     ERROR              ;ERROR
                                -2                      ;-----

5520                                     ;TAG PARITY GEN. & STORE TESTS
5521                                     ;READING CACHE MAINT. REGISTER
5522                                     ;BIT 9 FOR TAG PARITY DATA DID
5523                                     ;NOT RESULT IN 0.
5524 030064 000000                     0
5525 :::
5526 030066 000240             10$:  NOP                    ;END OF TEST
5527
5528
5529 030070 005237 001464             1157:  INC $TESIN          ;UPDATE TEST ID
:::
.SBTTL *****
.SBTTL TEST 157
.SBTTL
.SBTTL VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1 INTO TAG PARITY STORE
.SBTTL ADDRESS 0000 FOR FLOATING 1 PATTERN ON TAG PARITY GENERATOR INPUTS
.SBTTL
.SBTTL PROCEDURE: GENERATE 1 FROM TAG PARITY
.SBTTL GENERATOR BY PLACING FLOATING 1 PATTERN ON INPUTS
.SBTTL AND WRITING 1 INTO TAG PARITY STORE
.SBTTL LOCATION 0000.READ TAG PARITY STORE BIT FROM
.SBTTL CMR<9>
.SBTTL
.SBTTL CONDITIONS:
.SBTTL INPUTS TO TAG PARITY GEN:
.SBTTL TAG WRD<21:13> FLOATING 1 PATTERN
.SBTTL WWPD(1)- 0
.SBTTL TAG PARITY STORE ADDRESS:
.SBTTL CA<12:1>=0000
.SBTTL
.SBTTL RESULT: CMR<9> 1
.SBTTL *****
5549 030074 000004             SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                40$                ;ERROR/LOOP ON TEST
                                1$-40$+70000-14    ;TEST START LOCATION
                                0                  ;LOOP ON ERROR START LOCATION
                                25$-40$+70000-14    ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
                                40$:  MOV #OFF,CCR    ;LOOP ON ERROR END LOCATION
                                JSR R4,RELCTH      ;DISABLE CACHE
                                10$+2            ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
030076 030106
030100 070014
030102 000000
030104 070050
030106 012737 001015 177746
030114 004437 002352
030120 030244

```


5592
5593
5594
5595
5596
5597
5598
5599
5600

.....
.....
.....
.....
.....
.....
.....
.....
.....

INPUTS TO TAG PARITY GEN:
TAG WRD<21:13> ALL 0'S
WWPT(1)= 1
TAG PARITY STORE ADDRESS:
CA<12:1>=0000

RESULT: CMR<9>- 1

5601 030250 000004
030252 030262
030254 070004
030256 000000
030260 070040
030262 012737 001015 177746
030270 004437 002352
030274 030410

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$;ERROR/LOOP ON TEST
1\$-40\$+70000-14 ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTH ;DISABLE CACHE
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

5602 030276 005037 177752
5603 030302 112737 000003 177750
5604
5605
5606
5607
5608
5609 030310 012737 002015 177746
5610
5611 030316 005737 040000
5612 030322 005737 060000
5613
5614
5615 030326 005737 060000
5616
5617 030332 013701 177750
5618
030336 000240
030340 000240

1\$: CLR CHR ;LOAD AMR<8:0> WITH 0'S BY WRITING TO (CHR<8:0>
MOV B #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 INPUTS TO TAG PARITY GEN.
;TO BE LOADED FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE PARITY.
;ENABLE WRITE WRONG PARITY TAG
MOV #15+WWPT,CCR
TST 40000
TST 6000C ;PLACE ALL 0'S ON TAG WRD<20:13> INPUTS
;SINCE WWPT IS INVOKED A 1 WILL BE
;WRITTEN INTO PARITY STORE LOCATION 0000.
TST 60000 ;WRITE TAG BIT FROM
;LOCAT. 0000 INTO CMR<9>
MOV CMR,R1 ;SAVE CMR DATA
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
NOP ;FOR LOOP ON ERROR

5619 030342 012737 001015 177746
5620 030350 105037 177750
5621 030354 052737 000400 177746
5622 030362 032737 010000 177746
5623 030370 001374
5624 030372 032701 001000
5625 030376 001003
5626
030400 104000
030402 030400

MOV #OFF,CCR ;DISABLE CACHE
CLRB CMR ;DISABLE MAINT. MODE
BIS #FC,CCR ;BEFORE LEAVING TEST FLUSH CACHE
BIT #VCIP,CCR ;WAIT TILL DONE
BNE -6
BIT #TPB,R1 ;CHECK 1 FOR TAG PARITY STORE.
BNE 10\$;PASS
ERROR ;ERROR
.-2 ;-----

5627
5628
5629
5630

;TAG PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 9 FOR TAG PARITY DATA DID
;NOT RESULT IN 1.

5631 030404 000000
 5632
 5633 030406 000240
 5634
 5635
 5636
 5637 030410 005237 001464

```

0
::: 10$: NOP ;END OF TEST

T161: INC $TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 161
.SBTTL
.SBTTL WRITE AND READ 0'S TO ALL LOW CACHE DATA PARITY AND TAG PARITY STORES
.SBTTL
CONDITIONS:
:
: INPUTS TO DATA PARITY GEN:
: WRD<15:0> ALL 0'S
: WWPD(1)=0
: INPUTS TO TAG PARITY GEN.:
: TAG WRD<21:13> ALL 0'S
: WWPT(1)=0
: DATA PARITY/TAG PARITY STORE ADDRESS:
: CA<12:1>=0000 TO 3777

RESULT: CMR<11:9> ALL 0
:*****
    
```

5653 030414 000004
 030416 030426
 030420 070030
 030422 000000
 030424 070056
 030426 012737 001015 177746
 030434 004437 002352
 030440 030640

```

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40$ ;TEST START LOCATION
1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
40$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
10$+2 ;ADDRESS OF START OF NEXT TEST

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

5654 030442 005037 177752
 5655 030446 012705 060000
 5656 030452 005025
 5657 030454 020527 070000
 5658 030460 001374
 5659 030462 012705 060000
 5660 030466 012703 040000
 5661 030472 112737 000003 177750
 5662
 5663
 5664
 5665
 5666
 5667 030500 012737 000015 177746
 5668
 5669 030506 005713
 5670 030510 005715
 5671

```

CLR CHR ;LOAD AMR<8:0> WITH ALL 0'S
MOV #60000,R5 ;ADDRESS 60000 TO R5
2$: CLR (R5)+ ;CLEAR ALL LOW CACHE MAIN MEMORY
CMP R5,#70000
BNE 2$
MOV #60000,R5 ;1ST ADDRESS LOCATION IN R5
MOV #40000,R3 ;ADDRESS 40000 IN R3
1$: 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>
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
;
TST (R3)
TST (R5) ;WRITE 0'S INTO DATA/TAG PARITY STORE
;ADDRESS LOCATION SPECIFIED BY R5
    
```

```

5672 030512 005715          TST (R5)          ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
5673 030514 013701 177750  MOV CMR,R1       ;SAVE CMR DATA
5674          030520 000240          25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          030522 000240          NOP             ;FOR LOOP ON ERROR

5675 030524 105037 177750  CLRB CMR         ;DISABLE MAINT. MODE
5676 030530 012737 001015 177746  MOV #OFF,CCR     ;DISABLE CACHE
5677 030536 010537 047322  MOV R5,CA121    ;GET PARITY ADDRESS LOCATION USED
5678 030542 006237 047322  ASR CA121       ;PREPARE CA121 FOR TIMEOUT
5679 030546 032701 004000  BIT #HPB,R1     ;CHECK 0 HI BYTE PARITY STORE
5680 030552 001404  BEQ 7$          ;PASS
5681          030554 104000          ERROR           ;ERROR
          030556 030554          .-2              ;-----

5682          ;DATA/TAG PARITY GEN. & STORE TESTS
5683          ;READING CACHE MAINT. REGISTER
5684          ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5685          ;NOT RESULT IN 0.
5686 030560 047322  CA121          ;PRINT PARITY STORE ADDRESS LOCATION
5687          ;USED: CA<12:1>
5688 030562 000000  0
5689          ;:
5690 030564 032701 002000  ;:          7$: BIT #LPB,R1    ;CHECK 0 FOR LOWER BYTE PARITY DATA
5691 030570 001404  BEQ 8$          ;PASS
5692          ;
5693          030572 104000          ERROR           ;ERROR
          030574 030572          .-2              ;-----

5694          ;DATA/TAG PARITY GEN. & STORE TESTS
5695          ;READING CACHE MAINT. REGISTER
5696          ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5697          ;NOT RESULT IN 0.
5698 030576 047322  CA121          ;PRINT PARITY STORE ADDRESS USED:CA<12:1>
5699 030600 000000  0
5700          ;:
5701 030602 032701 001000  ;:          8$: BIT #TPB,R1    ;CHECK 0 FOR TAG PARITY DATA
5702 030606 001404  BEQ 9$          ;PASS
5703          030610 104000          ERROR           ;ERROR
          030612 030610          .-2              ;-----

5704          ;DATA/TAG PARITY GEN. AND STORAGE TESTS
5705          ;READING CACHE MAINT.REGISTER BIT 9 FOR
5706          ;TAG PARITY DATA DID NOT RESULT IN 0.
5707 030614 047322  CA121          ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
5708 030616 000000  0
5709          ;:
5710 030620 062705 000002  ;:          9$: ADD #2,R5      ;NEXT PARITY STORE ADDRESS LOCATION
5711 030624 062703 000002  ADD #2,R3
5712 030630 022705 070000  CMP #70000,R5   ;HAVE ALL LOW CACHE PARITY STORE ADDRES.
5713          ;LOCATIONS BEEN DONE
5714 030634 001316  BNE 1$          ;NO CONTINUE
5715 030636 000240  10$: NOP          ;END OF TEST
5716

```


5717
5718
5719
5720 030640 005237 001464

T162: INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 162
.SBTTL
.SBTTL WRITE AND READ 0'S TO ALL HI CACHE DATA PARITY AND TAG PARITY STORES
.SBTTL

5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735

CONDITIONS:
INPUTS TO DATA PARITY GEN:
WRTD<15:0> ALL 0'S
WWPD(1)=0
INPUTS TO TAG PARITY GEN.:
TAG WRTD<21:13> ALL 0'S
WWPT(1)=0
DATA PARITY/TAG PARITY STORE ADDRESS:
CA<12:1>=4000 TO 7777

RESULT: CMR<11:9> ALL 0

5736 030644 000004

:*****
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40\$;TEST START LOCATION
1\$-40\$+60000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25\$-40\$+60000-14 ;LOOP ON ERROR END LOCATION
40\$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
10\$+2 ;ADDRESS OF START OF NEXT TEST

030646 030656
030650 060030
030652 000000
030654 060056
030656 012737 001015 177746
030664 004437 002324
030670 031070

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

5737 030672 005037 177752
5738 030676 012705 070000
5739 030702 005025
5740 030704 020527 100000
5741 030710 001374
5742 030712 012705 070000
5743 030716 012703 050000
5744 030722 112737 000003 177750
5745
5746
5747
5748
5749
5750 030730 012737 000015 177746
5751
5752 030736 005713
5753 030740 005715
5754
5755 030742 005715
5756 030744 013701 177750
5757

CLR CHR ;LOAD AMR<8:0> WITH ALL 0'S
MOV #70000,R5 ;ADDRESS 70000 TO R5
2\$: CLR (R5)+ ;CLEAR ALL HI CACHE MAIN MEMORY
CMP R5,#100000
BNE 2\$
MOV #70000,R5 ;1ST ADDRESS LOCATION IN R5
MOV #5000,R3 ;ADDRESS 5000 IN R3
1\$: 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>
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
TST (R3)
TST (R5) ;WRITE 0'S INTO DATA/TAG PARITY STORE
;ADDRESS LOCATION SPECIFIED BY R5
TST (R5) ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
MOV CMR,R1 ;SAVE CMR DATA

```

030750 000240          25$:  NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
030752 000240          NOP          ;FOR LOOP ON ERROR

5758 030754 105037 177750 CLRB CMR          ;DISABLE MAINT. MODE
5759 030760 012737 001015 177746 MOV #OFF,CCR      ;DISABLE CACHE
5760 030766 010537 047322 MOV R5,CA121      ;GET PARITY ADDRESS LOCATION USED
5761 030772 006237 047322 ASR CA121          ;PREPARE CA121 FOR TYPEOUT
5762 030776 032701 004000 BIT #HPB,R1       ;CHECK 0 HI BYTE PARITY STORE
5763 031002 001404          BEQ 7$          ;PASS
5764
031004 104000          ERROR          ;ERROR
                                ;-----
031006 031004          .-2
5765
5766
5767
5768
5769 031010 047322          CA121
5770
5771 031012 000000          0
5772
5773 031014 032701 002000 :::          7$: BIT #LPB,R1      ;CHECK 0 FOR LOWER BYTE PARITY DATA
5774 031020 001404          BEQ 8$          ;PASS
5775
031022 104000          ERROR          ;ERROR
                                ;-----
031024 031022          .-2
5776
5777
5778
5779
5780 031026 047322          CA121
5781 031030 000000          0
5782
5783 031032 032701 001000 :::          8$: BIT #TPB,R1      ;CHECK 0 FOR TAG PARITY DATA
5784 031036 001404          BEQ 9$          ;PASS
5785
031040 104000          ERROR          ;ERROR
                                ;-----
031042 031040          .-2
5786
5787
5788
5789 031044 047322          CA121
5790 031046 000000          0
5791
5792 031050 062705 000002 :::          9$: ADD #2,R5          ;NEXT PARITY STORE ADDRESS LOCATION
5793 031054 062703 000002          ADD #2,R3
5794 031060 022705 100000          CMP #100000,R5      ;HAVE ALL HI CACHE PARITY STORE ADDRESS
5795
5796 031064 001316          BNE 1$          ;LOCATIONS BEEN DONE
5797 031066 000240          10$: NOP          ;NO,CONTINUE
5798
5799
5800
5801
5802
                                ;END OF TEST

```

5803 031070 005237 001464

T163: INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 163
.SBTTL
.SBTTL WRITE AND READ 1'S TO ALL LOW CACHE DATA PARITY AND TAG PARITY STORES
.SBTTL

5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818

CONDITIONS:
INPUTS TO DATA PARITY GEN:
WRD<15:0>= 000401
WWPD(1)= 0
INPUTS TO TAG PARITY GEN.:
TAG WRD<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

5819 031074 000004

:*****
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40\$;TEST START LOCATION
1\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25\$-40\$+70000-14 ;LOOP ON ERROR END LOCATION
40\$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
10.+2 ;ADDRESS OF START OF NEXT TEST

031076 031106
031100 070034
031102 000000
031104 070062
031106 012737 001015 177746
031114 004437 002352
031120 031324

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

5820 031122 012737 000001 177752
5821
5822 031130 012705 060000
5823 031134 012725 000401
5824
5825 031140 020527 070000
5826 031144 001373
5827 031146 012705 060000
5828 031152 012703 040000
5829 031156 112737 000003 177750
5830
5831
5832
5833
5834
5835 031164 012737 000015 177746
5836
5837 031172 005713
5838 031174 005715
5839
5840 031176 005715
5841 031200 013701 177750
5842
031204 000240

MOV #1,CHR ;LOAD AMR<8:0> WITH BIT PATTERN
;00000001
MOV #60000,R5 ;ADDRESS 60000 TO R5
2\$: MOV #401,(R5)+ ;WRITE A 401 IN ALL LOW CACHE MAIN
;MEMORY
CMP R5,#70000
BNF 2\$
MOV #60000,R5 ;1ST ADDRESS LOCATION IN R5
MOV #40000,R3 ;ADDRESS 40000 IN R3
1\$: MOVBS #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>
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
TST (R3)
TST (R5) ;WRITE 1'S INTO DATA/TAG PARITY STORE
;ADDRESS LOCATION SPECIFIED BY R5
TST (R5) ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
MOV CMR,R1 ;SAVE CMR DATA
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE

```

031206 000240          NOP          ;FOR LOOP ON ERROR
5843 031210 105037 177750          CLR# CMR          ;DISABLE MAINT. MODE
5844 031214 012737 001015 177746  MOV #OFF,CCR     ;DISABLE CACHE
5845 031222 010537 047322          MOV R5,CA121     ;GET PARITY ADDRESS LOCATION USED
5846 031226 006237 047322          ASR CA121        ;PREPARE CA121 FOR TYPEOUT
5847 031232 032701 004000          BIT #HPB,R1     ;CHECK 1 HI BYTE PARITY STORE
5848 031236 001004          BNE 7$         ;PASS
5849
031240 104000          ERROR          ;ERROR
                                ;-----
031242 031240          .-2
5850                                ;DATA/TAG PARITY GEN. & STORE TESTS
5851                                ;READING CACHE MAINT. REGISTER
5852                                ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5853                                ;NOT RESULT IN 1.
5854 031244 047322          CA121          ;PRINT PARITY STORE ADDRESS LOCATION
5855                                ;USED: CA<12:1>
5856 031246 000000          0
5857                                ;:::
5858 031250 032701 002000          7$: BIT #LPB,R1  ;CHECK 1 FOR LOWER BYTE PARITY DATA
5859 031254 001004          BNE 8$         ;PASS
5860
5861 031256 104000          ERROR          ;ERROR
                                ;-----
031260 031256          .-2
5862                                ;DATA/TAG PARITY GEN. & STORE TESTS
5863                                ;READING CACHE MAINT. REGISTER
5864                                ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5865                                ;NOT RESULT IN 1.
5866 031262 047322          CA121          ;PRINT PARITY STORE ADDRESS USED:CA<12:1>
5867 031264 000000          0
5868                                ;:::
5869 031266 032701 001000          8$: BIT #TPB,R1  ;CHECK 1 FOR TAG PARITY DATA
5870 031272 001004          BNE 9$         ;PASS
5871
031274 104000          ERROR          ;ERROR
                                ;-----
031276 031274          .-2
5872                                ;DATA/TAG PARITY GEN. AND STORAGE TESTS
5873                                ;READING CACHE MAINT.REGISTER BIT 9 FOR
5874                                ;TAG PARITY DATA DID NOT RESULT IN 1.
5875 031300 047322          CA121          ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
5876 031302 000000          0
5877                                ;:::
5878 031304 062705 000002          9$: ADD #2,R5    ;NEXT PARITY STORE ADDRESS LOCATION
5879 031310 062703 000002          ADD #2,R3
5880 031314 022705 070000          CMP #70000,R5  ;HAVE ALL LOW CACHE PARITY STORE ADDRESS
5881                                ;LOCATIONS BEEN DONE
5882 031320 001316          BNE 1$         ;NO,CONTINUE
5883 031322 000240          10$: NOP          ;END OF TEST
5884
5885
5886
5887
5888 031324 005237 001464          1164: INC $TESTN ;UPDATE TEST ID

```

5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903

```

:*****
.SBTTL
.SBTTL TEST 164
.SBTTL
.SBTTL WRITE AND READ 1'S TO ALL HI CACHE DATA PARITY AND TAG PARITY STORES
.SBTTL
CONDITIONS:
INPUTS TO DATA PARITY GEN:
WRD<15:0>= 000401
WDPD(1)= 0
INPUTS TO TAG PARITY GEN.:
TAG WRD<21:13> =BIT PATTERN 00000001
TWPT(1)=0
DATA PARITY/TAG PARITY STORE ADDRESS:
CA<12:1>=4000 TO 7777

RESULT: CMR<11:9> ALL 1'S

```

5904 031330 000004

031332 031342
031334 060034
031336 070000
031340 0c 062
031342 012737 001015 177746
031350 004437 002324
031354 031560

```

:*****
SCPCND ;SCOPE CONDIT'ONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40$ ;TEST START LOCATION
1$-40$+60000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25$-40$+60000-14 ;LOOP ON ERROR END LOCATION
40$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
10$+2 ;ADDRESS OF START OF NEXT TEST

```

```

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

```

5905 031356 012737 000001 177752
5906 031364 012705 070000
5907 031370 012725 000401
5908 031374 020527 100000
5909 031400 001373
5910 031402 012705 070000
5911 031406 012703 050000
5912 031412 112737 000003 177750
5913
5914
5915
5916
5917
5918 031420 012737 000015 177746
5919
5920 031426 005713
5921 031430 005715
5922
5923 031432 005715
5924 031434 013701 177750
5925
031440 000240
031442 000240
5926 031444 105037 177750

```

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$: MOVB #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY
; STORE BITS TO BE WRITTEN TO
;CMR<11:9> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;TDAR ALLOWS TAG PARITY STORE GENERATOR
;INPUTS TO BE LOADED FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
MOV #15,CCR
TST (R3)
TST (R5) ;WRITE 1'S INTO DATA/TAG PARITY STORE
;ADDRESS LOCATION SPECIFIED BY R5
TST (R5) ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
MOV CMR,R1 ;SAVE CMR DATA
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE

```

```

5927 031450 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
5928 031456 010537 047322      MOV R5,CA121     ;GET PARITY ADDRESS LOCATION USED
5929 031462 006237 047322      ASR CA121        ;PREPARE CA121 FOR TYPEOUT
5930 031466 032701 004000      BIT #HPB,R1     ;CHECK 1 HI BYTE PARITY STORE
5931 031472 001004                BNE 7$          ;PASS
5932 031474 104000                ERROR          ;ERROR
                    ;-----
                    ;-2
5933 031476 031474                ;DATA/TAG PARITY GEN. & STORE TESTS
5934                                ;READING CACHE MAINT. REGISTER
5935                                ;BIT 11 FOR UPPER BYTE PARITY DATA DID
5936                                ;NOT RESULT IN 1.
5937 031500 047322                CA121          ;PRINT PARITY STORE ADDRESS LOCATION
5938                                ;USED: CA<12:1>
5939 031502 000000                0
5940                                ;:::
5941 031504 032701 002000        7$: BIT #LPB,R1 ;CHECK 1 FOR LOWER BYTE PARITY DATA
5942 031510 001004                BNE 8$          ;PASS
5943 031512 104000                ERROR          ;ERROR
                    ;-----
                    ;-2
5944 031514 031512                ;DATA/TAG PARITY GEN. & STORE TESTS
5945                                ;READING CACHE MAINT. REGISTER
5946                                ;BIT 10 FOR LOWER BYTE PARITY DATA DID
5947                                ;NOT RESULT IN 1.
5948 031516 047322                CA121          ;PRINT PARITY STORE ADDRESS USED:CA<12:1>
5949 031520 000000                0
5950                                ;:::
5951 031522 032701 001000        8$: BIT #TPB,R1 ;CHECK 1 FOR TAG PARITY DATA
5952 031526 001004                BNE 9$          ;PASS
5953 031530 104000                ERROR          ;ERROR
                    ;-----
                    ;-2
5954 031532 031530                ;DATA/TAG PARITY GEN. AND STORAGE TESTS
5955                                ;READING CACHE MAINT.REGISTER BIT 9 FOR
5956                                ;TAG PARITY DATA DID NOT RESULT IN 0.
5957 031534 047322                CA121          ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
5958 031536 000000                0
5959                                ;:::
5960 031540 062705 000002        9$: ADD #2,R5   ;NEXT PARITY STORE ADDRESS LOCATION
5961 031544 062703 000002        ADD #2,R3
5962 031550 022705 100000        CMP #100000,R5 ;HAVE ALL HI CACHE PARITY STORE ADDRESS
5963                                ;LOCATIONS BEEN DONE
5964 031554 001716                BEQ 1$          ;NO CONTINUE
5965 031556 000240                10$: NOP       ;END OF TEST
5966
5967
5968
5969
5970
5971                                .SBTTL *
5972                                .SBTTL
5973                                .SBTTL *
5974                                .SBTTL

```

DATA/TAG PARITY STORE ADDRESS LINE TEST

5975
5976
5977
5978
5979
5980 031560 005237 001464

.SBTTL
.SBTTL *
T165: INC \$TESTN ;UPDATE TEST ID
:*****

5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995 031564 000004

.SBTTL
.SBTTL TEST 165
.SBTTL
.SBTTL
.SBTTL VERIFY HI & LO BYTE DATA PARITY STORE ADDRESS LINES
.SBTTL
: :
: : 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.
: :
: :*****

031566 031576
031570 070040
031572 000000
031574 070100
031576 012737 001015 177746
031604 004437 002352
031610 032034

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
40\$;TEST START LOCATION
1\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
25\$-40\$+70000-14 ;LOOP ON ERROR END LOCATION
40\$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
10\$+2 ;ADDRESS OF START OF NEXT TEST

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

5996 031612 000240
5997
5998
5999
6000 031614 005037 060000
6001 031620 012737 000002 047332
6002 031626 012702 040000
6003 031632 012703 060000
6004 031636 063702 047332
6005 031642 063703 047332
6006 031646 012713 000401
6007
6008 031652 112737 000002 177750
6009
6010
6011
6012 031660 012737 000015 177746
6013 031666 005737 040000
6014 031672 005737 060000
6015

NOP ;THIS 'NOP' WILL BE AT LOCATION 70000
;WHEN THE TEST IS RELOCATED TO HI
;CACHE. IT WILL BE OVERWRITTEN WITH
;'401' WHEN THE TEST IS EXECUTED.
CLR 60000 ;CLEAR LOCATION 60000 IN MAIN MEMORY
MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
2\$: 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
MOV #401,(R3) ;ODD DATA IN HI & LO BYTE AREAS OF
;LOCATION SPECIFIED BY R3
1\$: MOVB #HODO,CMR ;HODO ALLOWS HI & LO BYTE DATA PARITY
;STORE BITS TO BE WRITTEN TO CMR<11:10>
; ONLY DURING THE
;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
;NO UCB SO AS TO WRITE ENABLE DATA PARITY STORE
MOV #15,CCR
TST 40000
TST 60000
;READ UPDATE: WRITE 0'S INTO HI AND LO
;BYTE DATA PARITY STORES

6016	031676	005712			TST (R2)		
6017	031700	005713			TST (R3)		:WRITE 1 INTO HI & LO BYTE DATA
6018							:PARITY STORE LOCATION
6019							:SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
6020	031702	005737	060000		TST 60000		:LOAD DATA FROM HI & LO BYTE PARITY
6021							:DATA PARITY STORE LOCATION
6022							:0000 INTO CMR<11:10>.
6023	031706	013701	177750		MOV CMR,R1		:SAVE CMR DATA
6024							
	031712	000240		25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	031714	000240			NOP		:FOR LOOP ON ERROR
6025	031716	105037	177750		CLRB CMR		:DISABLE MAINT. MODE
6026	031722	012737	001015	177746	MOV #OFF,CCR		:DISABLE CACHE
6027	031730	032701	004000		BIT #HPB,R1		:READING CMR<11> FOR HI BYTE
6028							:DATA PARITY STORE DATA SHOULD RESULT
6029							:IN 0.
6030	031734	001411			BEQ 8\$:PASS
6031	031736	013737	047332	047322	MOV FLTPAT,CA121		:SAVE CA<12:1> USED
6032	031744	006237	047322		ASR CA121		:PREPARE CA121 FOR TYPEOUT
6033							
	031750	104000			ERROR		:ERROR
							:-----
	031752	031750			.-2		
6034							:HI & LO BYTE DATA PARITY STORE ADDRESS TEST
6035							:HI BYTE DATA PARITY STORE LOC. 0000
6036							:DID NOT READ AS A 0 INDICATING THAT
6037							:IT WAS OVERWRITTEN WITH A 1. THIS
6038							:SUGGESTS HI BYTE DATA PARITY STORE
6039							:ADDRESS LINE IS BAD.
6040	031754	047322			CA121		:PRINT PARITY STORE ADDRESS FAILURE
6041							:CA<12:1>.
6042							:NOTE THAT THE 1 IN THIS PATTERN
6043							:WILL POINT TO THE ADDRESS LINE OF
6044							:THAT BROUGHT OUT ERROR.
6045	031756	000000			0		
6046							
6047	031760	032701	004000	8\$:	BIT #HPB,R1		:READING CMR<10> FOR LO BYTE
6048							:DATA PARITY STORE DATA SHOULD RESULT
6049							:IN 0.
6050	031764	001411			BEQ 9\$:PASS
6051	031766	013737	047332	047322	MOV FLTPAT,CA121		:SAVE CA<12:1> USED
6052	031774	006237	047322		ASR CA121		:PREPARE CA121 FOR TYPEOUT
6053							
	032000	104000			ERROR		:ERROR
							:-----
	032002	032000			.-2		
6054							:HI & LO BYTE DATA PARITY STORE ADDRESS TEST
6055							:LO BYTE DATA PARITY STORE LOC. 0000
6056							:DID NOT READ AS A 0 INDICATING THAT
6057							:IT WAS OVERWRITTEN WITH A 1. THIS
6058							:SUGGESTS HI BYTE DATA PARITY STORE
6059							:ADDRESS LINE IS BAD.
6060	032004	047322			CA121		:PRINT PARITY STORE ADDRESS FAILURE
6061							:CA<12:1>.
6062							:NOTE THAT THE 1 IN THIS PATTERN
6063							:WILL POINT TO THE ADDRESS LINE OF


```

6064                                     ;THAT BROUGHT OUT ERROR.
6065 032006 000000                        0
6066
6067 032010 006337 047332                9$: ASL FLTPAT ;NEXT PATTERN
6068 032014 022737 020000 047332        CMP #20000,FLTPAT ;HAS DATA PARITY STORE ADDRESS 4000 BEEN D'NE?
6069 032022 001301                        BNE 2$ ;NO
6070 032024 012737 000240 070000        MOV #240,70000 ;RESTORE OVERWRITTEN LOCATION 70000
6071                                     ;WITH NOP.
6072 032032 000240                10$: NOP ;END OF TEST
6073
6074
6075
6076
6077 032034 005237 001464        T166: INC $TESTN ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 166
.SBTTL
.SBTTL
.SBTTL VERIFY TAG PARITY STORE ADDRESS LINES
.SBTTL
        PROCEDURE: WRITE 0 INTO TAG PARITY STORE
        ADDRESS LOCATION 0000.
        WRITE A 1 INTO TAG PARITY STORE
        ADDRESS LOCAT. 0001.
        READ TAG PARITY ADDRESS LOC.
        0000 FOR 0'S.
        REPEAT THE ABOVE SEQUENCE ,EACH TIME CHANGING THE
        ADDRESS LOCATION THE 1 IS WRITTEN INTO BY
        SHIFTING THE 1 ONE PLACE TO THE LEFT.
;*****
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092 032040 000004                SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        032042 032052                40$ ;TEST START LOCATION
        032044 070026                1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
        032046 000000                0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        032050 070066                25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
        032052 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
        032060 004437 002352        JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
        032064 032240                10$+2 ;ADDRESS OF START OF NEXT TEST

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

6093 032066 012737 000002 047332        MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
6094 032074 012702 040000                2$: MOV #40000,R2 ;ADDRESS 40000 INTO R2
6095 032100 012703 060000                MOV #60000,R3 ;ADRESS 60000 INTO R3
6096 032104 063702 047332                ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
6097 032110 063703 047332                ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
6098 032114 112737 000002 177750        1$: MOV# #HODO,CMR ;HODO ALLOWS TAG PARITY
6099                                     ;STORE BITS TO BE WRITTEN TO CMR<9>
6100                                     ; ONLY DURING THE
6101                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
6102 032122 012737 000015 177746        MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE DATA PARITY STORE
6103 032130 005737 040000                TST 40000
6104 032134 005737 060000                TST 60000 ;READ UPDATE: WRITE 0 INTO TAG

```

```

6105                                     ; PARITY STORE
6106 032140 005713                       TST (R3)
6107 032142 005712                       TST (R2)
6108                                     ;WRITE 1 INTO TAG
6109                                     ;PARITY STORE LOCATION
6110 032144 005737 060000                 TST 60000
6111                                     ;SPECIFIED BY R2'S BITS 1 THRU 12:CA<12:1>.
6112                                     ;LOAD DATA FROM
6113 032150 013701 177750                 MOV CMR,R1
6114                                     ;TAG PARITY STORE LOCATION
6114                                     ;0000 INTO CMR<9>.
6114 032154 000240 25$: NOP                ;SAVE CMR DATA
6114 032156 000240                       NOP
6114                                     ;INSTRUCTION 'JMP 1$' PLACED HERE
6114                                     ;FOR LOOP ON ERROR
6115 032160 105037 177750                 CLRB CMR
6116 032164 012737 001015 177746         MOV #OFF,CCR
6117 032172 032701 001000                 BIT #TPB,R1
6118                                     ;DISABLE MAINT. MODF
6119 032176 001411                         BEQ 9$
6120 032200 013737 047332 047322         MOV FLTPAT,CA121
6121 032206 006237 047322                 ASR CA121
6122                                     ;DISABLE CACHE
6122 032212 104000                         ERROR
6122                                     ;READING CMR<9> FOR TAG PARITY STORE
6122                                     ;DATA SHOULD RESULT IN 0
6122                                     ;PASS
6122                                     ;SAVE CA<12:1> USED
6122                                     ;PREPARE CA121 FOR TYPEOUT
6123 032214 032212                         .-2
6123                                     ;TAG PARITY STORE ADDRESS TEST
6124                                     ;TAG PARITY STORE LOC. 0000
6125                                     ;DID NOT READ AS A 0 INDICATING THAT
6126                                     ;IT WAS OVERWRITTEN WITH A 1. THIS
6127                                     ;SUGGESTS TAG PARITY STORE
6128                                     ;ADDRESS LINE IS BAD.
6129 032216 047322                         CA121
6129                                     ;PRINT PARITY STORE ADDRESS FAILURE
6130                                     ;CA<12:1>.
6131                                     ;NOTE THAT THE 1 IN THIS PATTERN
6132                                     ;WILL POINT TO THE ADDRESS LINE OF
6133                                     ;THAT BROUGHT OUT ERROR.
6134 032220 000000                         0
6135                                     ;:
6136 032222 006337 047332 9$: ASL FLTPAT
6137 032226 022737 020000 047332         CMP #20000,FLTPAT
6138 032234 001317                         BNE 2$
6139 032236 000240 10$: NOP                ;NEXT PATTERN
6139                                     ;HAS TAG PARITY STORE ADDRESS 4000 BEEN DONE?
6139                                     ;NO
6139                                     ;END OF TEST
6140
6141
6142
6143
6144 .SBTTL *
6145 .SBTTL *
6146 .SBTTL *
6147 .SBTTL *
6148 .SBTTL *
6149 .SBTTL *
6150
6151
6152
6153
6154
6155

```

DATA/TAG PARITY ERROR CHECKING

6156
6157 032240 005237 001464

T167: INC \$TESTN ;UPDATE TEST ID
:*****

6158
6159
6160
6161
6162
6163 032244 000004

.SBTTL
.SBTTL TES 167
.SBTTL
.SBTTL
.SBTTL VERIFY THAT ALL PARITY ERROR BITS IN CACHE MEMORY ERROR REGISTER
.SBTTL WILL READ 0 FOLLOWING A WRITE TO CME.
.SBTTL
:*****

032246 032256
032250 070000
032252 000000
032254 070016
032256 012737 001015 177746
032264 004437 002352
032270 032342

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$;ERROR/LOOP ON TEST
1\$-40\$+70000-14 ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40\$: MOV #OFF,CLR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTH ;DISABLE CACHE
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

6164 032272 112737 000002 177750
6165
6166
6167
6168
6169 032300 005037 177744
6170 032304 013701 177744
6171

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.

032310 000240
032312 000240

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

6172 032314 105037 177750
6173 032320 005701
6174 032322 001406
6175 032324 010137 047334
6176

CLRB CMR ;DISABLE MAINTENANCE MODE
TST R1 ;ARE ALL BITS 0?
BEQ 10\$;PASS
MOV R1,RECDAT ;GET CME CONTENTS RECEIVED

032330 104000
032332 032330

ERROR ;ERROR
.-2 ;-----

6177
6178
6179 032334 047334
6180 032336 000000

RECDAT ;PARITY ERROR CHECK TESTS
0 ;WRITING TO CME DID NOT LEAVE ALL PARITY ERROR BITS
0 ;PRINT CME CONTENTS RECEIVED

6181
6182 032340 000240
6183
6184
6185
6186
6187 032342 005237 001464

10\$: NOP ;END OF TEST
T170: INC \$TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 170

```

6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202 032346 CJU004
      032350 032360
      032352 070016
      032354 000000
      032356 070072
      032360 012737 001015 177746
      032366 004437 002352
      032372 032542

6203 032374 052737 000400 177746
6204 032402 032737 010000 177746
6205 032410 001374
6206 032412 005037 000000
6207 032416 112737 000002 177750
6208
6209
6210 032424 012737 000015 177746
6211 032432 005737 040000
6212 032436 005737 000000
6213
6214
6215 032442 005037 177744
6216 032446 005737 000000
6217
6218
6219
6220 032452 052737 000200 177746
6221
6222
6223 032460 000240
6224 032462 013701 177744
6225
      032466 000240
      032470 000240

6226 032472 012737 001015 177746
6227 032500 105037 177750
6228 032504 012737 000002 000000

```

```

.SBTTL
.SBTTL
.SBTTL VERIFY THAT CME CAN SHOW NO PARITY ERRORS FOLOWING A READ HIT
.SBTTL CONDITION
.SBTTL VERIFY TAG/DATA 'PARITY CHECK PARITY GENERATORS' WITH ALL 0'S
.SBTTL ON THEIR INPUTS.
.SBTTL
:
: 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
:
:*****

```

```

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40$ ;ERROR/LOOP ON TEST
1$-40$+70000-14 ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTH ;DISABLE CACHE
10$+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

```

```

1$: BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
BIT ~VCIP,CCR ;WAIT TILL DONE
BNE -6
CLR 0 ;0'S TO MAIN MEMORY LOCATION 0
MOVB #HODO,CMR ;HODO ALLOWS CLOCKING OF PARITY INFO TO
;CME ONLY DURING THE DESTINATION ACCESS OF
;OF AN INSTRUCTION
;NO UCB SO AS TO WRITE CACHE STORES
;READ UPDATE TO CACHE LOCATION 0000-
;ALL 0'S WILL BE WRITTEN INTO DATA/TAG STORES
;AND A 0 INTO DATA/TAG PARITY STORES
CLR CME ;CLEAR CME
TST 0 ;READ HIT-ALL 0'S WILL BE PLACED ON INPUTS
;OF DATA/TAG PARITY DETECT PARITY GENERATORS
;AND ALL PARITY INFO WILL BE CLOCKED
;TO CME
BIS #PEA,CCR ;SET CCR<7> SO AS TO ENABLE CME<7>,<6>,<5> TO
;TO BE WRITTEN INDIVIDUALLY FROM
;PARITY INFO LOGIC,AND TO WRITE CME<15>.
NOP
MOV CME,R1 ;SAVE CME CONTENTS
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
MOV #OFF,CCR ;DISABLE CACHE
CLRB CMR ;DISABLE MAINTENANCE MODE
MOV #2,0 ;RESTORE LOCATION C

```

6229 032512 005701
 6230 032514 001411
 6231
 6232 032516 005037 047320
 6233 032522 010137 047334
 6234
 6235
 032526 104000
 032530 032526
 6236
 6237
 6238 032532 047320
 6239 032534 047334
 6240
 6241 032536 000000
 6242
 6243 032540 000240
 6244
 6245
 6246

TST R1
 BEQ 10\$
 CLR EXDAT6
 MOV R1,RECDAT

;WERE ALL PARITY ERROR BITS IN CME-0?
 ;PASS;NEXT TEST
 ;SPECIFY CME CONTENTS EXPECTED
 ;GET CME CONTENTS RECEIVED

ERROR
 .-2

;ERROR
 ;-----

EXDAT6
 RECDAT

;PARITY CHECK TESTS
 ;ALL PARITY ERROR BITS IN CME SHOULD HAVE READ 0
 ;PRINT EXPECTED CME CONTENTS
 ;PRINT CONTENTS OF CME RECEIVED

0

:::

*0\$: NOP

;END OF TEST

6248
6249 032542 005237 001464

T171: INC \$TESTN ;UPDATE TEST ID
:*****

6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279 032546 000004

.SBTTL
.SBTTL TEST 171
.SBTTL
.SBTTL
.SBTTL VERIFY THE FOLLOWING WHEN A LOCATION PREVIOUSLY WRITTEN
.SBTTL WITH WRONG TAG PARITY IS ACCESSED:
.SBTTL
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
.SBTTL
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
:*****

032550 032560
032552 070016
032554 000000
032556 070126
032560 012737 001015 177746
032566 004437 002352
032572 033100

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$;ERROR/LOOP ON TEST
1\$-40\$+70000-14 ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTH ;DISABLE CACHE
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

6280 032574 052737 000400 177746
6281 032602 032737 010000 177746
6282 032610 001374
6283 032612 005037 000000
6284 032616 112737 000002 177750
6285
6286
6287 032624 012737 000015 177746
6288 032632 005737 000000

2\$: BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
BIT #VCIP,CCR ;WAIT TILL DONE
BNE -6
1\$: CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
MOVB #HODO,CMR ;HODO ALLOWS CLOCKING OF PARITY INFO TO
;CME ONLY DURING THE DESTINATION ACCESS OF
;AN INSTRUCTION.
MOV #15,CCR ;NO JCB SO AS TO WRITE CACHE STORES
TST 0
;

```

6289 032636 005737 040000          TST 40000          ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STOR
6290 032642 052737 002000 177746  BIS #WWPT,CCR     ;ALLOW WRITE WRONG PARITY TO TAGG PARITY STORE
6291 032650 005737 000000          TST 0              ;READ UPDATE TO CACHE LOCATION 0000;
6292                                     ;ALL 0'S(EVEN DATA) WILL BE WRITTEN TO DATA/TAG STOR
6293                                     ;0'S INTO DATA PARITY STORES,AND A 1
6294                                     ;INTO TAG PARITY STORE.
6295 032654 042737 002000 177746  BIC #WWPT,CCR     ;DISABLE WWPT
6296 032662 005037 177744          CLR CME           ;CLEAR CME
6297 032666 005737 000000          TST 0              ;READ HIT; ALL 0'S WILL BE PLACED ON INPUTS
6298                                     ;OF DATA PARITY ERROR CHECK PARITY GEN'S, BUT
6299                                     ;TAG PARITY CHECK GENERATOR WILL
6300                                     ;SEE ODD DATA DUE TO WRONG PARITY
6301                                     ;FROM PREVIOUS READ UPDATE.
6302 032672 013700 177744          MOV CME,RO        ;SAVE PARITY ERROR BITS WITH PEA CLEARED
6303 032676 052737 000200 177746  BIS #PEA,CCR      ;SET CCR<7> SO AS TO WRITE CME<7>,<6>,<5> INDIVIDUAL
6304                                     ;FROM PARITY CHECK LOGIC, AND TO WRITE CME<15>
6305
6306 032704 000240          NOP
6307 032706 013701 177744          MOV CME,R1        ;SAVE CME CONTENTS AFTER PEA IS SET.
6308                                     ;LOOKS LIKE EXPECTED TRAP
6309                                     ;DID NOT OCCUR.
6310 032712 005037 177744          CLR CME           ;CLEAR CME
6311 032716 013702 177744          MOV CME,R2        ;SAVE CME CONTENTS AFTER CLEAR
6312
        032722 000240          NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
        032724 000240          NOP                ;FOR LOOP ON ERROR
6313 032726 012737 001015 177746  MOV #OFF,CCR      ;DISABLE CACHE
6314 032734 105037 177750          CLR B CMR        ;DISABLE MAINT. MODE
6315 032740 012737 000002 000000  MOV #2,0          ;RESTORE LOCATION 0
6316 032746 052737 000400 177746  BIS #FC,CCR       ;BEFORE LEAVING TEST,FLUSH CACHE TO
6317                                     ;ELIMINATE ANY EFFECTS OF WWPT
6318 032754 032737 010000 177746  BIT #VCIP,CCR     ;WAIT TILL DONE
6319 032762 001374          BNE .-6
6320 032764 022700 000340          CMP #340,RO      ;WERE PARITY ERROR BITS CORRECT IN CME
6321                                     ;WITH PEA CLEARED?
6322 032770 001412          BEQ 7$           ;YES
6323 032772 012737 000340 047320  MOV #340,EXDAT6  ;SPECIFY CME CONTENTS EXPECTED
6324 033000 010037 047334          MOV RO,RECDAT   ;GET CME CONTENTS RECEIVED
6325
        033004 104000          ERROR                ;ERROR
        033006 033004          .-2                ;-----
6326
6327                                     ;PERITY CHECK TESTS
6328                                     ;PARITY ERROR BITS DID NOT READ CORRECTLY
6329 033010 047320          EXDAT6            ;WITH PEA CLEARED
6330 033012 047334          RECDAT          ;PRINT CME CONTENTS EXPECTED
6331 033014 000000          0                ;PRINT CME CONTENTS RECEIVED
6332
        033016 022701 100040          CMP #100040,R1   ;WERE PARITY ERROR BITS CORRECT WITH PEA SET?
6333 033022 001412          BEQ 8$           ;YES
6334 033024 012737 100040 047320  MOV #100040,EXDAT6 ;SPECIFY CME CONTENTS EXPECTED
6335 033032 010137 047334          MOV R1,RECDAT   ;GET CME CONTENTS RECEIVED
6336
        033036 104000          ERROR                ;ERROR
        ;-----

```

```

033040 033036      .-2
6338          ;PARITY CHECK TESTS
6339          ;PARITY ERROR BITS DID NOT READ CORRECTLY
6340          ;WITH PEA SET.
6341 033042 047320  EXDAT6      ;PRINT CME CONTENTS EXPECTED
6342 033044 047334  RECDAT      ;PRINT CME CONTENTS RECEIVED
6343 033046 000000  0
6344
6345 033050 005702      8$:  TST R2      ;DID CME CLEAR?
6346 033052 001411      BEQ 10$    ;YES
6347 033054 005037 047320 CLR EXDAT6 ;SPECIFY CME CONTENTS EXPECTED
6348 033060 010237 047334 MOV R2,RECDAT ;GET CME CONTENTS RECEIVED
6349
033064 104000      ERROR          ;ERROR
                                ;-----
033066 033064      .-2
6350          ;PARITY CHECK TESTS
6351          ;PARITY ERROR BITS DID NOT CLEAR
6352          ;FOLLOWING WRITE TO CME
6353 033070 047320  EXDAT6      ;PRINT CME CONTENTS EXPECTED
6354 033072 047334  RECDAT      ;PRINT CME CONTENTS RECEIVED
6355 033074 000000  0
6356
6357 033076 000240      10$:  NOP          ;END OF TEST
6358
6359
6360
6361 033100 005237 001464 T172:      INC $TESTN      ;UPDATE TEST ID

```

```

:*****
.SBTTL
.SBTTL TEST 172
.SBTTL
.SBTTL
.SBTTL VERIFY THE FOLLOWING WHEN A LOCATION PREVIOUSLY WRITTEN
.SBTTL WITH WRONG LO & HI BYTE PARITY IS ACCESSED.
.SBTTL
.SBTTL 1. A PARITY ERROR IS DETECTED AND ALL PARITY ERROR BITS IN
.SBTTL CME READ CORRECTLY WITH PEA CLEARED.
.SBTTL 2. ALL PARITY ERROR BITS READ CORRECTLY WITH PEA
.SBTTL SET.
.SBTTL 3. A WRITE TO CME CLEARS CME<15> ,<7> AND <6> FROM A 1 STATE
.SBTTL
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

```

6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386

6387
6388
6389
6390
6391
6392
6393
6394

.....
.....
.....
.....
.....
.....
.....

PEA SET:
CME<15>=1
CME<7>,<6>=1
CME<5>=0

033104 000004

033106 033116
033110 070016
033112 000000
033114 070126
033116 012737 001015 177746
033124 004437 002352
033130 033436

SCPCND
40\$
1\$-40\$+70000-14
0
25\$-40\$+70000-14
40\$: MOV #OFF,CCR
JSR R4,RELCTH
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

6395 033132 052737 000400 177746
6396 033140 032737 010000 177746
6397 033146 001374
6398 033150 005037 000000
6399
6400 033154 112737 000002 177750
6401
6402
6403 033162 012737 000015 177746
6404 033170 005737 000000
6405 033174 005737 040000
6406 033200 052737 000100 177746
6407
6408 033206 005737 000000
6409
6410
6411
6412 033212 042737 000100 177746
6413 033220 005037 177744
6414 033224 005737 000000
6415
6416
6417
6418
6419
6420 033230 013700 177744
6421 033234 052737 000200 177746
6422
6423
6424 033242 000240
6425 033244 013701 177744
6426
6427
6428 033250 005037 177744
6429 033254 013702 177744
6430
033260 000240

BIS #FC,CCR
BIT #VCIP,CCR
BNE .-6
1\$: CLR 0

MOV B #HODO,CMR

MOV #15,CCR
TST 0
TST 4000
BIS #WWPD,CCR

TST 0

BIC #WWPD,CCR
CLR CME
TST 0

MOV CME,R0
BIS #PEA,CCR

NOP
MOV CME,R1

CLR CME
MOV CME,R2

25\$: NOP

: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 STORE
: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
:INTO TAG PARITY STORE.
:DISABLE WWPD
:CLEAR CME
:READ HIT; ALL 0'S(EVEN DATA) WILL BE
:PLACED ON INPUTS
:OF TAG PARITY ERROR CHECK PARITY GEN'S, BUT
:LO & HI BYTE PARITY CHECK GENERATORS WILL
:SEE ODD DATA DUE TO WRONG PARITY
:FROM PREVIOUS READ UPDATE.
:SAVE PARITY ERROR BITS WITH PEA CLEARED
:SET CCR<7> SO AS TO WRITE CME<7>,<6>,<5> INDIVIDUAL
:FROM PARITY CHECK LOGIC, AND TO WRITE CME<15>

:SAVE CME CONTENTS AFTER PEA IS SET.
:LOOKS LIKE EXPECTED TRAP
:DID NOT OCCUR.
:CLEAR CME
:SAVE CME CONTENTS AFTER CLEAR

:INSTRUCTION 'JMP 1\$' PLACED HERE

```

033262 000240      NOP      :FOR LOOP ON ERROR
6431 033264 012737 001015 177746  MOV #OFF,CCR      :DISABLE CACHE
6432 033272 105037 177750      CLR B CMB         :DISABLE MAINT. MODE
6433 033276 012737 000002 000000  MOV #2,0          :RESTORE LOCATION 0
6434 033304 052737 000400 177746  BIS #FC,CCR       :BEFORE LEAVING TEST FLUSH CACHE TO
6435                                     :ELIMINATE ANY EFFECTS OF WWPB
6436 033312 032737 010000 177746  BIT #VCIP,CCR     :
6437 033314 001374      BNE #-6           :
6438 033322 022700 000340      CMP #340,R0       :WERE PARITY ERROR BITS CORRECT IN CME
6439                                     :WITH PEA CLEARED?
6440 033326 001412      BEQ 7$           :YES
6441 033330 012737 000340 047320  MOV #340,EXDAT6  :SPECIFY CME CONTENTS EXPECTED
6442 033336 010037 047334      MOV R0,RECDAT    :GET CME CONTENTS RECEIVED
6443
033342 104000      ERROR      :ERROR
033344 033342      .-2      :
6444                                     :PARITY CHECK TESTS
6445                                     :PARITY ERROR BITS DID NOT REAL CORRECTLY
6446                                     :WITH PEA CLEARED
6447 033346 047320      EXDAT6         :PRINT CME CONTENTS EXPECTED
6448 033350 047334      RECDAT         :PRINT CME CONTENTS RECEIVED
6449 033352 000000      0
6450
6451 033354 022701 100300      7$:  CMP #100300,R1    :WERE PARITY ERROR BITS CORRECT WITH PEA SET?
6452 033360 001412      BEQ 8$           :YES
6453 033362 012737 100300 047320  MOV #100300,EXDAT6 :SPECIFY CME CONTENTS EXPECTED
6454 033370 010137 047334      MOV R1,RECDAT    :GET CME CONTENTS RECEIVED
6455
033374 104000      ERROR      :ERROR
033376 033374      .-2      :
6456                                     :PARITY CHECK TESTS
6457                                     :PARITY ERROR BITS DID NOT REAL CORRECTLY
6458                                     :WITH PEA SET.
6459 033400 047320      EXDAT6         :PRINT CME CONTENTS EXPECTED
6460 033402 047334      RECDAT         :PRINT CME CONTENTS RECEIVED
6461 033404 000000      0
6462
6463 033406 005702      8$:  TST R2           :DID CME CLEAR?
6464 033410 001411      BEQ 10$          :YES
6465 033412 005037 047320      CLR EXDAT6       :SPECIFY CME CONTENTS EXPECTED
6466 033416 010237 047334      MOV R2,RECDAT   :GET CME CONTENTS RECEIVED
6467
033422 104000      ERROR      :ERROR
033424 033422      .-2      :
6468                                     :PARITY CHECK TESTS
6469                                     :PARITY ERROR BITS DID NOT CLEAR
6470                                     :FOLLOWING WRITE TO CME
6471 033426 047320      EXDAT6         :PRINT CME CONTENTS EXPECTED
6472 033430 047334      RECDAT         :PRINT CME CONTENTS RECEIVED
6473 033432 000000      0
6474
6475 033434 000240      10$:  NOP      :END OF TEST
6476

```

6477
6478
6479

```

6481 .SBTTL *
6482 .SBTTL
6483 .SBTTL *
6484 .SBTTL
6485 .SBTTL
6486 .SBTTL *
6487
6488
6489
6490
6491
6492 033436 005237 001464 *173: INC $TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 173
.SBTTL
.SBTTL
.SBTTL VERIFY INTERRUPT LOGIC BY ASSURING THAT A TRAP OCCURS TO LOCATION
.SBTTL 114 WHEN A LOCATION PREVIOUSLY WRITTEN
.SBTTL WITH WRONG HI/LO BYTE PARITY IS ACCESSED.
.SBTTL
:
:
: CONDITIONS: PEA-0
: DCPI-0
:*****
6503 033442 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
033444 033454 40$ ;TEST START LOCATION
033446 070016 1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
033450 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
033452 070152 25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
033454 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
033462 004437 002352 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
033466 033726 10$+2 ;ADDRESS OF START OF NEXT TEST

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

6504 033470 052737 000400 177746 BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
6505 033476 032737 010000 177746 BIT #VCIP,CCR ;WAIT TILL DONE
6506 033504 001374 BNE -6
6507 033506 005037 000000 1$: CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
6508 033512 012737 070142 000114 MOV #4$-40$+67764,114 ;SETUP FOR CACHE TRAP
6509 033520 012737 000340 000116 MOV #340,116
6510
6511 033526 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES
6512 ;AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC
6513 ; ONLY DURING THE DESTINATION ACCESS OF
6514 ;AN INSTRUCTION.
6515 033534 005037 002060 CLR FAIL1 ;CLEAR ERROR FLAG
6516 033540 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6517 033546 005737 000000 TST 0
6518 033552 005737 040000 TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORE
6519 033556 052737 000100 177746 BIS #WVDP,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO
6520 ;& HI BYTE PARITY STORE.
6521 033564 005737 000000 TST 0 ;READ UPDATE TO CACHE LOCATION 0000;

```



```

6611                                     ;ELIMINATE ANY EFFECTS OF WWPD
6612 034170 032737 010000 177746      BIT #VCIP,CCR      ;WAIT TILL DONE
6613 034176 001374                                     BNE .-6
6614 034200 005737 002060              TST FAIL1        ;DID TRAP OCCUR?
6615 034204 001403                      BEQ 10$          ;NO
6616                                     ERROR              ;ERROR
                                           ;-----
                                           .-2
6617                                     ; INTERRUPT/ABORT LOGIC TESTS
6618                                     ; TRAP TO LOCATION 114 OCCURED
6619                                     ; AND WAS NOT INHIBITED BY DCPI
6620
6621 034212 000000                      0
6622
6623 034214 000240              10$:  NOP              ;END OF TEST      :::
6624
6625
6626
6627
6628 034216 005237 001464      T175:          INC $TESTN          ;UPDATE TEST ID
:::*****
.SBTTL
.SBTTL TEST 175
.SBTTL
.SBTTL
.SBTTL
.SBTTL VERIFY ABORT LOGIC BY THE FOLLOWING RESULTS WHEN A LOCATION
.SBTTL PREVIOUSLY WRITTEN WITH WRONG HI/LO BYTE PARITY IS ACCESSED.
.SBTTL 1. CME<15> WILL SET CAUSED BY ABORT SIGNAL BEING ASSERTED
.SBTTL 2. WRITE TO CME WILL CLEAR CME<15>
.SBTTL 3. INSTRUCTION CYCLE WILL BE ABORTED
.SBTTL 4. THE ABORT CAUSES TRAP TO 114
.SBTTL
.SBTTL
.SBTTL PROCEDURE:          INHIBIT CLOCKING OF PARITY ERROR SIGNAL TO
.SBTTL                    INTERRUPT LOGIC. ALLOW CME<15> TO BE SET
.SBTTL                    BY ABORT SIGNAL WHICH IS ASSERTED BY PARITY
.SBTTL                    ERROR SIGNAL TO ABORT LOGIC.
.SBTTL
.SBTTL
.SBTTL
.SBTTL CONDITIONS:      PEA=1
.SBTTL                    DCPI=1
.SBTTL
.SBTTL *****
6629
6630
6631
6632
6633
6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647 034222 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                           ;ERROR/LOOP ON TEST
                                           ;TEST START LOCATION
034224 034234              40$              ;LOOP ON ERROR START LOCATION
034226 070016              1$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
034230 000000              0              ;LOOP ON ERROR END LOCATION
034232 070174              25$-40$+70000-14 ;DISABLE CACHE
034234 012737 001015 177746      40$:  MOV #OFF,CCR    ;LOCATE TEST CODE TO HIGH CACHE SPACE
034242 004437 002352          JSR R4,RELCTH    ;ADDRESS OF START OF NEXT TEST
034246 034574              10$+2
                                           ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                           ;ARE RELOCATED TO HI CACHE SPACE
6648 034250 052737 000400 177746      BIS #FC,CCR      ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS

```

6649	034256	032737	010000	177746		BIT #VCIP,CCR	:WAIT TILL DONE
6650	034264	001374				BNE .-6	
6651	034266	005037	000000		1\$:	CLR 0	:ALL 0'S TO LOCATION 0
6652	034272	005000				CLR R0	:ADDRESS 0 TO R0
6653	034274	012737	070150	000114		MOV #4\$-40\$+67764,114	:SETUP FOR TRAP
6654	034302	012737	000340	000116		MOV #340,116	
6655							
6656	034310	112737	000002	177750		MOV B #HODO,CMR	:HODO ALLOWS CACHE UPDATES
6657							:AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC
6658							: ONLY DURING THE DESTINATION ACCESS OF
6659							:AN INSTRUCTION.
6660	034316	005037	002060			CLR FAIL1	:CLEAR ERROR FLAG
6661	034322	012703	177777			MOV #-1,R3	:ALL 1'S TO R3
6662	034326	012737	000015	177746		MOV #15,CCR	:NO UCB SO AS TO WRITE CACHE STORES
6663	034334	005710				TST (R0)	
6664	034336	005737	040000			TST 40000	:UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STOR
6665	034342	052737	000100	177746		BIS #WVDP,CCR	:AL OW WRITE WRONG PARITY DATA TO LO
6666							: & HI BYTE PARITY STORE.
6667	034350	005710				TST (R0)	:READ UPDATE TO CACHE LOCATION 0000
6668							:WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES
6669	034352	042737	000100	177746		BIC #WVDP,CCR	:DISABLE WVDP
6670	034360	005037	177744			CLR CME	:CLEAR CME AND PARITY DETECT LOGIC
6671	034364	042737	000004	177746		BIC #FMLO,CCR	:ENABLE LOW CACHE
6672	034372	052737	000200	177746		BIS #PEA,CCR	:ALLOW FOR ABORT
6673	034400	011003				MOV (R0),R3	:READ HIT;
6674							:LO & HI BYTE PARITY CHECK GENERATORS WILL
6675							: DETECT WRONG PARITY
6676							:USING HODO AND SOURCE MODE FOR READING
6677							:LOCATION 0 WILL INHIBIT PARITY ERROR
6678							:FROM BEING CLOCKED TO INTERRUPT LOGIC
6679							:HOWEVER, THE PARITY ERROR SIGNAL
6680							:WILL CAUSE THE ABORT SIGNAL TO BE
6681							:ASSERTED.THE ABORT SIGNAL WILL BE
6682							:CAUSE CME<15> TO BE SET.
6683							:THIS INSTRUCTION SHOULD BE ABORTED
6684	034402	000240				NOP	
6685	034404	005237	002060			INC FAIL1	:INDICATE NO TRAP OCCURED
6686	034410	012737	001015	177746		MOV #OFF,CCR	:DISABLE CACHE
6687	034416	000404				BR 5\$	
6688	034420	012737	001015	177746	4\$:	MOV #OFF,CCR	:DISABLE CACHE
6689	034426	022626				CMP (SP)+,(SP)+	:READJUST STACK
6690	034430	013701	177744		5\$:	MOV CME,R1	:SAVE CME CONTENTS
6691	034434	005037	177744			CLR CME	:WRITE TO CME
6692	034440	013702	177744			MOV CME,R2	:SAVE CME CONTENTS
6693							
6694							
	034444	000240			25\$:	NOP	:INSTRUCTION 'JMP 1\$' PLACED HERE
	034446	000240				NOP	:FOR LOOP ON ERROR
6695	034450	105037	177750			CLRB CMR	:DISABLE MAINT. MODE
6696	034454	012737	000002	000000		MOV #2,@#0	:RESTORE VECTORS
6697	034462	012737	000116	000114		MOV #116,@#114	
6698	034470	005037	000116			CLR @#116	
6699	034474	052737	000400	177746		BIS #FC,CCR	:BEFORE LEAVING TEST FLUSH CACHE TO
6700							:ELIMINATE ANY EFFECTS OF WVDP
6701	034502	032737	010000	177746		BIT #VCIP,CCR	:WAIT TILL DONE
6702	034510	001374				BNE .-6	


```

6703 034512 032701 100000 BIT #CMPE,R1 ;WAS CME<15> SET?
6704 034516 001003 BNE 7$ ;YES; PASS
6705 034520 104000 ERROR ;ERROR
;-----
034522 034520 .-2 ;INTERRUPT/ABORT LOGIC
;CME<15> WAS NOT SET DUE TO ABORT
;SIGNAL
6706 0
6707
6708
6709 034524 000000 0
6710
6711 034526 032702 100000 7$: BIT #CMPE,R2 ;:::
6712 034532 001403 BEQ 8$ ;WAS CME<15> CLEARED?
6713 034534 104000 ERROR ;ERROR
;-----
034536 034534 .-2 ;INTERRUPT/ABORT TESTS
;CME<15> WAS NOT CLEARED BY WRITE TO
; TO CME
6714 0
6715
6716
6717 034540 000000 0
6718
6719 034542 022703 177777 8$: CMP #-1,R3 ;:::
6720 034546 001403 BEQ 9$ ;WAS INSTRUCTION ABORTED LEAVING R3 INTACT?
6721 034550 104000 ERROR ;ERROR
;-----
034552 034550 .-2 ;INTERRUPT/ABORT TESTS
;R3 WAS OVERWRITTEN WITH DATA INDICATING
; THAT INSTRUCTION WAS NOT ABORTED
6722 0
6723
6724
6725 034554 000000 002060 9$: TST FAIL1 ;DID TRAP OCCUR
6726 034556 005737 BEQ 10$ ;YES;PASS
6727 034562 001403 ERROR ;ERROR
;-----
034564 104000 .-2 ;INTERRUPT/ABORT TESTS
;TRAP DID NOT OCCUR DUE TO ABORT
6728 034566 034564 0
6729
6730
6731 034570 000000 0
6732
6733 034572 000240 10$: NOP ;END OF TEST
6734
6735
6736
6737
6738 .SBTTL *
6739 .SBTTL *
6740 .SBTTL * CACHE TAG/DATA PARITY STORE RAM MEMORY MARCH PATTERN TEST
6741 .SBTTL *
6742 .SBTTL *
6743
6744
6745
6746 034574 005237 001464 1176: INC $TESTN ;UPDATE TEST ID
;*****

```

6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762

6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788

034600 000004

034602 034612
034604 070000
034606 000000
034610 070126
034612 012737 001015 177746
034620 004437 002352
034624 035226

012702 000401
005037 177752
012700 060000
005020
020027 070000
001374
012700 060000
012701 040000
112737 000003 177750

012737 000015 177746

005721
005720

022700 070000
001373

012737 000001 177752
012700 060000
042737 000001 177750

.SBTTL
.SBTTL TEST 176
.SBTTL
.SBTTL
.SBTTL VERIFY CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S BY PERFORMING A
.SBTTL 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.

.....
SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$:ERROR/LOOP ON TEST
1\$-40\$+70000-14 :TEST START LOCATION
0 :LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40\$: MOV #OFF,CCR :LOOP ON ERROR END LOCATION
JSR R4,RELCTH :DISABLE CACHE
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

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
MOVB #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>
MOV #15,CCR :NO UCB SO AS TO WRITE ENABLE PARITY
 :STORE
6\$: TST (R1)+
TST (R0)+ :WRITE 0'S INTO ALL DATA/TAG PARITY STORE
 :ADDRESS LOCATIONS SPECIFIED BY R0
CMP #70000,R0 :DONE?
BNE 6\$:NO
7\$: MOV #1,CHR :LOAD AMR<8:0> BY WRITING TO CHR<8:0>
MOV #60000,R0 :ADDR. 60000 TO RC
BIC #TDAR,CMR :DISABLE TDAR TO ALLOW UPDATE
 :OF CACHE TAG STORE THRU CA<21:13>

6789	034730	005710		TST (R0)	:READ MISS TO CACHE LOCATION SPECIFIED
6790					:BY R0. CLOCK TAG/DATA PARITY STORE
6791					:BITS INTO CMR<11:9>. SHOULD BE ALL 0'S.
6792					:ALSO CAUSES UPDATE TO CACHE. TAG/DATA
6793					:PARITY STORE LOCATION SHOULD REMAIN
6794					:WITH 0'S.
6795	034732	013705	177750	MOV CMR,R5	:SAVE CMR CONTENTS
6796	034736	052737	000001 177750	BIS #TDAR,CMR	:ENABLE TDAR TO ALLOW TAG PARITY GENERATOR
6797					:INPUTS TO SEE ODD DATA FROM AMR<8:0>
6798	034744	010210		MOV R2,(R0)	:WRITE HIT CAUSES UPDATE TO CACHE.
6799					:TAG/DATA PARITY STORES WILL BE WRITTEN
6800					:WITH 1'S DUE TO AMR<8:0> ODD DATA
6801					:AND PATTERN 401 FROM R2 BEING PUT
6802					:ONTO PAX DATA LINES RESULTING IN
6803					:ODD DATA FOR LO AND HI BYTE DATA PARITY
6804					:GENERATORS.
6805					
6806	034746	005710		TST (R0)	:READ MISS. CLOCK TAG/DATA PARITY STORE BITS TO
6807					:CMR<11:9>. SHOULD BE ALL 1'S.
6808	034750	013703	177750	MOV CMR,R3	:SAVE CMR CONTENTS
6809					
	034754	000240		25\$: NOP	:INSTRUCTION 'JMP 1\$' PLACED HERE
	034756	000240		NOF	:FOR LOOP ON ERROR
6810					
6811	034760	042705	170777	BIC #170777,R5	:INTERESTED IN BITS 11:9
6812	034764	005705		TST R5	:BITS 11:9 SHOULD BE ALL 0'S
6813	034766	001437		BEQ 8\$:PASS
6814					
6815	034770	012737	001015 177746	MOV #OFF,CCR	:DISABLE CACHE
6816	034776	105037	177750	CLRB CMR	:CLEAR MAINT. MODE
6817	035002	005037	047324	CLR EXDAT1	:SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
6818	035006	010537	047342	MOV R5,CMR119	:SPECIFY CACHE TAG/DATA PARITY STORE DATA RECEIVED
6819					:THRU CMR<11:9>
6820	035012	012737	000011 002052	MOV #9,LOOP	:PREPARE CMR119 FOR TYPEOUT
6821	035020	006237	047342	ASR CMR119	
6822	035024	042737	100000 047342	BIC #100000,CMR119	
6823	035032	005337	002052	DEC LOOP	
6824	035036	001370		BNE 11\$	
6825					
6826	035040	010037	047322	MOV R0,CA121	:SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATI
6827	035044	006237	047322	ASR CA121	
6828					
	035050	104000		ERROR	:ERROR
					:-----
	035052	035050		.-2	
6829					:TAG/DATA PARITY STORE MARCH PATTERN TEST
6830					:READING CACHE TAG/DATA PARITY STORE DATA
6831					:THRU CMR<11:9> DID NOT READ ALL 0'S.
6832					:THIS SUGGESTS THAT A RAM LOCATION
6833					:SPECIFIED BY CA121 WAS OVERWRITTEN
6834					:WITH A 1 WHEN WRITING A 1 TO ANOTHER
6835					:LOCATION. ANY BIT IN CMR119 DATA
6836					:THAT IS A 1 MAY POINT TO A BAD
6837					:CACHE TAG/DATA PARITY STORE RAM.
6838					
6839	035054	047324		EXDAT1	:PRINT EXPECTED CACHE TAG/DATA PARITY STORE DATA

```
6840 035056 047342          CMR119          ;PRINT CACHE TAG/DATA PARITY STORE RECEIVED
6841                          ;THRU CMR<11:9>
6842 035060 047322          CA121          ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATI
6843 035062 000000          0
6844 035064 000452          BR 3$         ;END THE TEST
6845 035066 042703 170777 8$: BIC #170777,R3 ;INTERESTED IN BITS 11:9 ONLY
6846 035072 022703 007000   CMP #7000,R3  ;BITS 11:9 SHOULD BE ALL 1'S
6847 035076 001440          BEQ 9$        ;PASS
6848
6849 035100 012737 001015 177746 MOV #OFF,CCR  ;DISABLE CACHE
6850 035106 105037 177750   CLRB CMR     ;CLEAR MAINT. MODE
6851 035112 012737 000007 047324 MOV #7,EXDAT1 ;SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
6852 035120 010337 047342   MOV R3,CMR119 ;SPECIFY CACHE TAG/DATA PARITY STORE DATA READ
6853
6854 035124 012737 000011 002052 MOV #9,LOOP   ;THRU CMR<11:9>
6855 035132 006237 047342 12$: ASR CMR119   ;PREPARE CMR119 FOR TYPEOUT
6856 035136 042737 100000 047342 BIC #100000,CMR119
6857 035144 005337 002052   DEC LOOP
6858 035150 001370          BNE 12$
6859 035152 010037 047322   MOV R0,CA121 ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATI
6860 035156 006237 047322   ASR CA121
6861
035162 104000          ERROR          ;ERROR
035164 035162          .-2          ;-----
6862
6863                          ;TAG/DATA PARITY STORE MARCH PATTERN TEST
6864                          ;READING CACHE TAG/DATA PARITY STORE DATA
6865                          ;THRU CMR<11:9> DID NOT READ ALL 1'S.
6866                          ;ANY BIT IN CMR119 DATA
6867                          ;THAT IS A 0 MAY POINT TO A BAD
6868                          ;CACHE TAG/DATA PARITY STORE RAM.
6869 035166 047324          EXDAT1
6870 035170 047342          CMR119
6871
6872 035172 047322          CA121
6873 035174 000000          0
6874 035176 000405          BR 3$         ;END TEST
6875 035200 062700 000002 9$: ADD #2,R0    ;NEXT LOCATION
6876 035204 022700 070000   CMP #70000,R0 ;HAS ALL LO CACHE BEEN DONE?
6877 035210 001244          BNE 7$        ;NO,CONTINUE
6878 035212 012737 001015 177746 3$: MOV #OFF,CCR ;DISABLE CACHE
6879 035220 105037 177750   CLRB CMR     ;DISABLE MAINT. MODE
6880 035224 000240          10$: NOP      ;END OF TEST
6881
6882
6883
6884 035226 005237 001464 T177: INC $TESTN ;UPDATE TEST ID
;*****
;SBTTL
;SBTTL TEST 177
;SBTTL
;SBTTL
;SBTTL VERIFY CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S BY PERFORMING A
;SBTTL MARCH PATTERN TEST TO HIGH CACHE AREA OF TAG/DATA PARITY STORE(LOC. 4000-7777)
6885 ;:
6886 ;: PROCEDURE: 1. WRITE ALL 0'S TO ALL HI CACHE TAG/DATA PARITY STORE
6887 ;: RAMS CORRESPONDING TO LOCATIONS 4000-7777
6888 ;:
6889 ;:
```

6890
6891
6892
6893
6894
6895
6896
6897
6898
6899

:::
:::
:::
:::
:::
:::
:::
:::
:::
:::
:::

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.

6900 035232 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

035234 035244
 035236 060000
 035240 000000
 035242 060126
 035244 012737 001015 177746
 035252 004437 002324
 035256 035660

40\$
 1\$-40\$+60000-14
 0
 25\$-40\$+60000-14
 40\$: MOV #OFF,CCR
 JSR R4,RELCTL
 10\$+2

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

6901 035260 012702 000401
 6902 035264 005037 177752
 6903 035270 012700 070000
 6904 035274 005020
 6905 035276 020027 100000
 6906 035302 001374
 6907 035304 012700 070000
 6908 035310 012701 050000
 6909 035314 112737 000003 177750
 6910
 6911
 6912
 6913
 6914
 6915 035322 012737 000015 177746
 6916
 6917 035330 005721
 6918 035332 005720
 6919
 6920 035334 022700 100000
 6921 035340 001373
 6922
 6923 035342 012737 000001 177752
 6924 035350 012700 070000
 6925 035354 042737 000001 177750
 6926
 6927 035362 005710
 6928
 6929
 6930
 6931
 6932
 6933 035364 013705 177750
 6934 035370 052737 000001 177750

1\$: MOV #401,R2
 CLR CHR
 MOV #70000,R0
 2\$: CLR (R0)+
 CMP R0,#100000
 BNE 2\$
 MOV #70000,R0
 MOV #50000,R1
 MOVB #HODO+TDAR,CMR
 MOV #15,CCR
 6\$: TST (R1)+
 TST (R0)+
 CMP #100000,R0
 BNE 6\$
 MOV #1,CHR
 MOV #70000,R0
 7\$: BIC #TDAR,CMR
 TST (R0)
 MOV CMR,R5
 BIS #TDAR,CMR

:SETUP R2 WITH PATTERN 401
 :LOAD AMR<8:0> WITH ALL 0'S
 :ADDRESS 70000 TO R0
 :CLEAR ALL HIGH CACHE MAIN MEMORY
 :1ST ADDRESS LOCATION IN R0
 :ADDRESS 50000 IN R1
 :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
 :WRITE 0'S INTO ALL DATA/TAG PARITY STORE
 :ADDRESS LOCATIONS SPECIFIED BY R0
 :DONE?
 :NO
 :LOAD AMR<8:0> BY WRITING TO CHR<8:0>
 :ADDR. 70000 TO R0
 :DISABLE TDAR TO ALLOW UPDATE
 :OF CACHE TAG STORE THRU CA<21:13>
 :READ MISS TO CACHE LOCATION SPECIFIED
 :BY R0. CLOCK TAG/DATA PARITY STORE
 :BITS INTO CMR<11:9>. SHOULD BE ALL 0'S.
 :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

```

6935
6936 035376 010210          MOV R2,(R0)
6937
6938
6939
6940
6941
6942
6943
6944 035400 005710          TST (R0)
6945
6946 035402 013703 177750    MOV CMR,R3
6947
        035406 000240          25$: NOP
        035410 000240          NOP
6948
6949 035412 042705 170777    BIC #170777,R5
6950 035416 005705          TST R5
6951 035420 001437          BEQ 8$
6952
6953 035422 012737 001015 177746    MOV #OFF,CCR
6954 035430 105037 177750    CLR B CMR
6955 035434 005037 047324    CLR EXDAT1
6956 035440 010537 047342    MOV R5,CMR119
6957
6958 035444 012737 000011 002052    MOV #9,LOOP
6959 035452 006237 047342          11$: ASR CMR119
6960 035456 042737 100000 047342    BIC #100000,CMR119
6961 035464 005337 002052    DEC LOOP
6962 035470 001370          BNE 11$
6963
6964 035472 010037 047322    MOV R0,CA121
6965 035476 006237 047322    ASR CA121
6966
        035502 104000          ERROR
        035504 035502          .-2
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977 035506 047324          EXDAT1
6978 035510 047342          CMR119
6979
6980 035512 047322          CA121
6981 035514 000000          0
6982 035516 000452          BR 3$
6983 035520 042703 170777          8$: BIC #170777,R3
6984 035524 022703 007000    CMP #7000,R3
6985 035530 001440          BEQ 9$
; 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 TYPLOUT
; SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATI
; 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 LOCATI
; END THE TEST
; INTERESTED IN BITS 11:9 ONLY
; BITS 11:9 SHOULD BE ALL 1'S
; PASS

```

```

6986
6987 035532 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
6988 035540 105037 177750      CLRB CMR          ;CLEAR MAINT. MODE
6989 035544 012737 000007 047324      MOV #7,EXDAT1    ;SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
6990 035552 010337 047342      MOV R3,CMR119    ;SPECIFY CACHE TAG/DATA PARITY STORE DATA READ
6991                                     ;THRU CMR<11:9>
6992 035556 012737 000011 002052      MOV #9,LOOP      ;PREPARE CMR119 FOR TYPEOUT
6993 035564 006237 047342      12$: ASR CMR119
6994 035570 042737 100000 047342      BIC #100000,CMR119
6995 035576 005337 002052      DEC LOOP
6996 035602 001370
6997 035604 010037 047322      MOV R0,CA121     ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATI
6998 035610 006237 047322      ASR CA121
6999
      035614 104000      ERROR           ;ERROR
      035616 035614      .-2            ;-----
7000                                     ;TAG/DATA PARITY STORE MARCH PATTERN TEST
7001                                     ;READING CACHE TAG/DATA PARITY STORE DATA
7002                                     ;THRU CMR<11:9> DID NOT READ ALL 1'S.
7003                                     ;ANY BIT IN CMR119 DATA
7004                                     ;THAT IS A 0 MAY POINT TO A BAD
7005                                     ;CACHE TAG/DATA PARITY STORE RAM.
7006
7007 035620 047324      EXDAT1          ;PRINT EXPECTED CACHE TAG/DATA PARITY STORE DATA
7008 035622 047342      CMR119         ;PRINT CACHE TAG/DATA PARITY STORE DATA READ
7009                                     ;THRU CMR<11:9>
7010 035624 047322      CA121          ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATI
7011 035626 000000      0
7012 035630 000405      BR 3$
7013 035632 062700 000002 9$: ADD #2,R0      ;END TEST
7014 035636 022700 100000      CMP #100000,R0  ;NEXT LOCATION
7015 035642 001244      BNE 7$         ;HAS ALL HI CACHE BEEN DONE?
7016 035644 012737 001015 177746      3$: MOV #OFF,CCR ;NO,CONTINUE
7017 035652 105037 177750      CLRB CMR       ;DISABLE CACHE
7018 035656 000240      10$: NOP       ;DISABLE MAINT. MODE
7019                                     ;END OF TEST
7020
7021
7022
7023
7024
7025      .SBTTL *
7026      .SBTTL
7027      .SBTTL *      CACHE VALID STORE RAM MEMORY MARCH PATTERN TEST -SET A
7028      .SBTTL
7029      .SBTTL *
7030
7031
7032
7033 035660 005237 001464      T200:          INC $TESTN      ;UPDATE TEST ID
      ;*****
      .SBTTL
      .SBTTL TEST 200
      .SBTTL
      .SBTTL
7034      .SBTTL
7035      .SBTTL VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A

```

7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046

.SBITL 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.

7048
7049

7050 035664 000004

035666 035676
035670 070026
035672 000000
035674 070136
035676 012737 001015 177746
035704 004437 002352
035710 036214

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\$
1\$-40\$+70000-14
0
25\$-40\$+70000-14
40\$: MOV #OFF,CCR
JSR R4,RELCTH
10\$+2

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

7051 035712 032737 020000 177746
7052 035720 001407
7053 035722 052737 000400 177746
7054 035730 032737 010000 177746
7055 035736 001374

BIT #VSIU,CCR
BEQ 1\$
BIS #FC,CCR
BIT #VCIP,CCR
BNE .-6

:IS SET A USED
:YES
:FLUSH

7056
7057 035740
7058 035740 012700 060000
7059 035744 012701 040000
7060 035750 005002
7061 035752 112737 000002 177750

1\$:

MOV #60000,R0
MOV #40000,R1
CLR R2
MOV B #HODO,CMR

:1ST ADDRESS LOCATION IN R0
:ADDRESS 40000 IN R1
:HODO ALLOWS CACHE UPDATES & VALID
:STORE BITS TO BE WRITTEN TO
:CMR<12> ONLY DURING THE DESTINATION
:ACCESS OF AN INSTRUCTION.
:STORE

7062
7063
7064
7065
7066 035760 012737 000015 177746
7067 035766 005721
7068 035770 005720
7069
7070 035772 022700 070000
7071 035776 001373

6\$:

MOV #15,CCR
TST (R1)+
TST (R0)+
CMP #70000,R0
BNE 6\$

:NO UCB SO AS TO UPDATE VALID STORE
:UPDATE ALL LO CACHE VALID STORE
:ADDRESS LOCATIONS SPECIFIED BY R0
:DONE?
:NO

7072
7073 036000 012700 060000
7074 036004 052737 001000 177746
7075 036012 010220
7076
7077 036014 022700 070000
7078 036020 001374
7079 036022 042737 001000 177746

13\$:

MOV #60000,R0
BIS #UCB,CCR
MOV R2,(R0)+
CMP #70000,R0
BNE 13\$
BIC #UCB,CCR

:ADDR. 60000 TO R0
:ENABLE UCB
:WRITE HIT WITH UCB WILL INVALIDATE
:OR WRITE 0 TO ALL LO CACHE VALID STORE
:DONE?
:NO
:DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES

7080
7081 036030 012700 060000
7082 036034
7083 036034 005710
7084
7085
7086
7087
7088
7089 036036 013705 177750
7090 036042 005710
7091
7092 036044 013703 177750

7\$:

MOV #60000,R0
TST (R0)
MOV CMR,R5
TST (R0)
MOV CMR,R3

:ADDRESS 60000 TO R0
:READ MISS TO CACHE LOCATION SPECIFIED
: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

```

7093      036050 000240      25$:  NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
          036052 000240      NOP      ;FOR LOOP ON ERROR

7094
7095 036054 042705 167777      BIC #167777,R5      ;INTERESTED IN BIT 12
7096 036060 005705      TST R5      ;BIT 12 SHOULD BE 0
7097 036062 001416      BEQ 8$      ;PASS
7098
7099 036064 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
7100 036072 105037 177750      CLR B CMR      ;CLEAR MAINT. MODE
7101 036076 010037 047322      MOV R0,CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7102 036102 006237 047322      ASR CA121
7103      036106 104000      ERROR      ;ERROR
          ;-----
          036110 036106      .-2

7104      ;VALID STORE MARCH PATTERN TEST- SET A
7105      ;READING CACHE VALID STORE DATA
7106      ;THRU CMR<12> DID NOT READ 0.
7107      ;THIS SUGGESTS THAT A RAM LOCATION
7108      ;SPECIFIED BY CA121 WAS OVERWRITTEN
7109      ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
7110      ;LOCATION.
7111      ;THIS INDICATES THAT VALID STORE RAM
7112      ;SET A IS BAD.
7113
7114 036112 047322      CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7115 036114 000000      0
7116 036116 000430      BR 3$      ;END THE TEST
7117 036120 042703 167777      8$:  BIC #167777,R3      ;INTERESTED IN BIT 12 ONLY
7118 036124 022703 010000      CMP #10000,R3      ;BIT 12 SHOULD BE 1
7119 036130 001416      BEQ 9$      ;PASS
7120
7121 036132 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
7122 036140 105037 177750      CLR B CMR      ;CLEAR MAINT. MODE
7123 036144 010037 047322      MOV R0,CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7124 036150 006237 047322      ASR CA121
7125      036154 104000      ERROR      ;ERROR
          ;-----
          036156 036154      .-2

7126      ;VALID STORE MARCH PATTERN TEST- SET A
7127      ;READING CACHE VALID STORE DATA
7128      ;THRU CMR<12> DID NOT READ 1.
7129      ;THIS SUGGESTS THAT VALID STORE RAM
7130      ;IC SET A IS BAD.
7131
7132      ;THRU CMR<12>
7133 036160 047322      CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7134 036162 000000      0
7135 036164 000405      BR 3$      ;END TEST
7136 036166 062700 000002      9$:  ADD #2,R0      ;NEXT LOCATION
7137 036172 022700 070000      CMP #70000,R0      ;HAS ALL LO CACHE BEEN DONE?
7138 036176 001316      BNE 7$      ;NO,CONTINUE
7139 036200 012737 001015 177746      3$:  MOV #OFF,CCR      ;DISABLE CACHE
7140 036206 105037 177750      CLR B CMR      ;DISABLE MAINT. MODE

```

7141 036212 000240 10\$: NOP ;END OF TEST

7142
7143
7144
7145

7146 036214 005237 001464 T201: INC \$TESTN ;UPDATE TEST ID

.SBTTL
.SBTTL TEST 201
.SBTTL

7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161

.SBTTL VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
.SBTTL 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.

7162 036220 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
036222 036232 40\$;TEST START LOCATION
036224 060026 1\$-40\$+60000-14 ;LOOP ON ERROR START LOCATION
036226 000000 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
036230 060136 25\$-40\$+60000-14 ;LOOP ON ERROR END LOCATION
036232 012737 001015 177746 40\$ MOV #OFF,CCR ;DISABLE CACHE
036240 004437 002324 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
036244 036550 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

7163 036246 032737 020000 177746 BIT #VSIU,CCR ;IS SET A USED
7164 036254 001407 BEQ 1\$;YES
7165 036256 052737 000400 177746 BIS #FC,CCR ;FLUSH
7166 036264 032737 010000 177746 BIT #VCIP,CCR
7167 036272 001374 BNE -6

7168
7169 036274 1\$: MOV #70000,R0 ;1ST ADDRESS LOCATION IN R0
7170 036274 012700 070000 MOV #50000,R1 ;ADDRESS 50000 IN R1
7171 036300 012701 050000 CLR R2
7172 036304 005002 MOV# #HODO,CMR

7173 036306 112737 000002 177750 ;HODO ALLOWS CACHE UPDATES & VALID
7174 ; STORE BITS TO BE WRITTEN TO
7175 ;CMR<12> ONLY DURING THE DESTINATION
7176 ;ACCESS OF AN INSTRUCTION.
7177 ;STORE

7178 036314 012737 000015 177746 6\$: MOV #15,CCR ;NO UCB SO AS TO UPDATE VALID STORE
7179 036322 005721 TST (R1)+
7180 036324 005720 TST (R0)+
7181 ;UPDATE ALL HI CACHE VALID STORE
;ADDRESS LOCATIONS SPECIFIED BY R0

```

7182 036326 022700 100000      CMP #100000,R0      ;DONE?
7183 036332 001373              BNE 6$             ;NO
7184                               ;
7185 036334 012700 070000      MOV #70000,R0      ;ADDR. 70000 TO R0
7186 036340 052737 001000 177746 13$: BIS #UCB,CCR       ;ENABLE UCB
7187 036346 010220              MOV R2,(R0)+       ;WRITE HIT WITH UCB WILL INVALIDATE
7188                               ;OR WRITE 0 TO ALL HI CACHE VALID STORE
7189 036350 022700 100000      CMP #100000,R0     ;DONE?
7190 036354 001374              BNE 13$           ;NO
7191 036356 042737 001000 177746 7$: BIC #UCB,CCR       ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
7192                               ;
7193 036364 012700 070000      MOV #70000,R0      ;ADDRESS 70000 TO R0
7194 036370              ;
7195 036370 005710              TST (R0)          ;READ MISS TO CACHE LOCATION SPECIFIED
7196                               ;BY R0. CLOCK VALID STORE
7197                               ;BIT INTO CMR<12>. SHOULD BE 0.
7198                               ;ALSO CAUSES UPDATE TO CACHE.
7199                               ;VALID STORE LOCATION WILL BE WRITTEN
7200                               ;WITH A 1.
7201 036372 013705 177750      MOV CMR,R5         ;SAVE CMR CONTENTS
7202 036376 005710              TST (R0)          ;READ HIT. CLOCK VALID STORE BIT TO CMR<12>
7203                               ;SHOULD BE 1.
7204 036400 013703 177750      MOV CMR,R3         ;SAVE CMR CONTENTS
7205                               ;
7206                               ;INSTRUCTION 'JMP 1$' PLACED HERE
7207 036404 000240 25$: NOP      ;FOR LOOP ON ERROR
7208 036406 000240              NOP
7209                               ;
7210                               ;INTERESTED IN BIT 12
7211 036410 042715 167777      BIC #167777,R5    ;BIT 12 SHOULD BE 0
7212 036414 005705              TST R5            ;PASS
7213 036416 001416              BEQ 8$
7214                               ;
7215                               ;DISABLE CACHE
7216 036420 012737 001015 177746 8$: MOV #OFF,CCR       ;CLEAR MAINT. MODE
7217 036426 105037 177750      CLR# CMR          ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7218 036432 010037 047322      MOV R0,CA121
7219 036436 006237 47322      ASR CA121
7220                               ;
7221 036442 104000              ERROR              ;ERROR
7222                               ;-----
7223 036444 036442              .-2
7224                               ;
7225                               ;VALID STORE MARCH PATTERN TEST- SET A
7226 036446 047322              CA121             ;READING CACHE VALID STORE DATA
7227 036450 000000              0                ;THRU CMR<12> DID NOT READ 0.
7228 036452 000430              BR 3$            ;THIS SUGGESTS THAT A RAM LOCATION
7229 036454 042703 167777      8$: BIC #167777,R3 ;SPECIFIED BY CA121 WAS OVERWRITTEN
7230 036460 022703 010000      CMP #10000,R3     ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
7231 036464 001416              BEQ 9$           ;LOCATION.
7232                               ;THIS INDICATES THAT VALID STORE RAM
7233                               ;SET A IS BAD.
7234                               ;
7235                               ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7236                               ;
7237                               ;END THE TEST
7238                               ;INTERESTED IN BIT 12 ONLY
7239                               ;BIT 12 SHOULD BE 1
7240                               ;PASS

```

```

7233 036466 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
7234 036474 105037 177750      CLRB CMR          ;CLEAR MAINT. MODE
7235 036500 010037 047322      MOV R0,CA121     ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7236 036504 006237 047322      ASR CA121
7237                                ERROR                ;ERROR
                                :-----
                                .-2
7238                                ;VALID STORE MARCH PATTERN TEST- SET A
7239                                ;READING CACHE VALID STORE DATA
7240                                ;THRU CMR<12> DID NOT READ 1.
7241                                ;THIS SUGGESTS THAT VALID STORE RAM
7242                                ;IC SET A IS BAD.
7243
7244                                ;THRU CMR<12>
7245 036514 047322      CA121            ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7246 036516 000000      0
7247 036520 000405      BR 3$           ;END TEST
7248 036522 062700 000002 9$:    ADD #2,R0        ;NEXT LOCATION
7249 036526 022700 100000      CMP #100000,R0  ;HAS ALL HI CACHE BEEN DONE?
7250 036532 001316      BNE 7$         ;NO,CONTINUE
7251 036534 012737 001015 177746 3$:    MOV #OFF,CCR    ;DISABLE CACHE
7252 036542 105037 177750      CLRB CMR        ;DISABLE MAINT. MODE
7253 036546 000240      10$:          NOP          ;END OF TEST
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267 036550 005237 001464      T202:          INC $TESTN      ;UPDATE TEST ID

```

```

.SBTTL *
.SBTTL
.SBTTL *          CACHE VALID STORE RAM MEMORY MARCH PATTERN TEST -SET B
.SBTTL
.SBTTL *

```

```

7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282

```

T202: INC \$TESTN ;UPDATE TEST ID
:*****
.SBTTL
.SBTTL TEST 202
.SBTTL
.SBTTL
.SBTTL VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
.SBTTL 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.
:*****

7283	036554	006004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	036556	036566			40\$:ERROR/LOOP ON TEST
	036560	070026			1\$-40\$+70000-14		:TEST START LOCATION
	036562	000000			0		:LOOP ON ERROR START LOCATION
	036564	070136			25\$-40\$+70000-14		:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	036566	012737	001015	177746	40\$: MOV #OFF,CCR		:LOOP ON ERROR END LOCATION
	036574	004437	002352		1SR R4,RELCTH		:DISABLE CACHE
	036600	037104			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
7284	036602	032737	02000	177746	BIT #VSIU,CCR		:IS SET B USED
7285	036610	001007			BNE 1\$:YES
7286	036612	052737	000400	177746	BIS #FC,CCR		:FLUSH
7287	036620	032737	010000	177746	BIT #VCIP,CCR		
7288	036626	001374			BNE .-6		
7289							
7290	036630				1\$:		
7291	036630	012700	060000		MOV #60000,R0		:1ST ADDRESS LOCATION IN R0
7292	036634	012701	040000		MOV #40000,R1		:ADDRESS 40000 IN R1
7293	036640	005002			CLR R2		
7294	036642	112737	000002	177750	MOV B #HODO,CMR		:HODO ALLOWS CACHE UPDATES & VALID
7295							:STORE BITS TO BE WRITTEN TO
7296							:CMR<12> ONLY DURING THE DESTINATION
7297							:ACCESS OF AN INSTRUCTION.
7298							:STORE
7299	036650	012737	000015	177746	MOV #15,CCR		:NO UCB SO AS TO UPDATE VALID STORE
7300	036656	005721			6\$: TST (R1)+		:
7301	036660	005720			TST (R0)+		:UPDATE ALL LO CACHE VALID STORE
7302							:ADDRESS LOCATIONS SPECIFIED BY R0
7303	036662	022700	070000		CMP #70000,R0		:DONE?
7304	036666	001373			BNE 6\$:NO
7305							
7306	036670	012700	060000		MOV #60000,R0		:ADDR. 60000 TO R0
7307	036674	052737	001000	177746	BIS #UCB,CCR		:ENABLE UCB
7308	036702	010220			13\$: MOV R2,(R0)+		:WRITE HIT WITH UCB WILL INVALIDATE
7309							:OR WRITE 0 TO ALL LO CACHE VALID STORE
7310	036704	022700	070000		CMP #70000,R0		:DONE?
7311	036710	001374			BNE 13\$:NO
7312	036712	042737	001000	177746	BIC #UCB,CCR		:DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
7313							
7314	036720	012700	060000		2\$: MOV #60000,R0		:ADDRESS 60000 TO R0
7315	036724						
7316	036724	005710			TST (R0)		:READ MISS TO CACHE LOCATION SPECIFIED
7317							:BY R0. CLOCK VALID STORE
7318							:BIT INTO CMR<12>. SHOULD BE 0.
7319							:ALSO CAUSES UPDATE TO CACHE.
7320							:VALID STORE LOCATION WILL BE WRITTEN
7321							:WITH A 1.
7322	036726	013705	177750		MOV CMR,R5		:SAVE CMR CONTENTS
7323	036732	005710			TST (R0)		:READ HIT.CLOCK VALID STORE BIT TO CMR<12>
7324							:SHOULD BE 1.
7325	036734	013703	177750		MOV CMR,R3		:SAVE CMR CONTENTS
7326							
	036740	000240			25\$: NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE

```

036742 000240          NOP          ;FOR LOOP ON ERROR
7327
7328 036744 042705 167777      BIC #167777,R5      ;INTERESTED IN BIT 12
7329 036750 005705              TST R5              ;BIT 12 SHOULD BE 0
7330 036752 001416              BEQ 8$              ;PASS
7331
7332 036754 012737 001015 177746 MOV #OFF,CCR        ;DISABLE CACHE
7333 036762 105037 177750      CLRB CMR            ;CLEAR MAINT. MODE
7334 036766 010037 047322      MOV R0,CA121        ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7335 036772 006237 047322      ASR CA121
7336
036776 104000          ERROR          ;ERROR
                                ;-----
037000 036776          .-2
7337
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347 037002 047322              CA121              ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7348 037004 000000              0
7349 037006 000430              BR 3$              ;END THE TEST
7350 037010 042703 167777      8$: BIC #167777,R3      ;INTERESTED IN BIT 12 ONLY
7351 037014 022703 010000      CMP #10000,R3      ;BIT 12 SHOULD BE 1
7352 037020 001416              BEQ 9$              ;PASS
7353
7354 037022 012737 001015 177746 MOV #OFF,CCR        ;DISABLE CACHE
7355 037030 105037 177750      CLRB CMR            ;CLEAR MAINT. MODE
7356 037034 010037 047322      MOV R0,CA121        ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7357 037040 006237 047322      ASR CA121
7358
037044 104000          ERROR          ;ERROR
                                ;-----
037046 037044          .-2
7359
7360
7361
7362
7363
7364
7365
7365 037050 047322              CA121              ;THRU CMR<12>
7366 037052 000000              0                  ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7367 037054 000405              BR 3$              ;END TEST
7368 037056 062700 000002      9$: ADD #2,R0        ;NEXT LOCATION
7369 037062 022700 070000      CMP #70000,R0      ;HAS ALL LO CACHE BEEN DONE?
7370 037066 001316              BNE 7$              ;NO, CONTINUE
7371 037070 012737 001015 177746 MOV #OFF,CCR        ;DISABLE CACHE
7372 037076 105037 177750      CLRB CMR            ;DISABLE MAINT. MODE
7373 037102 000240          10$: NOP              ;END OF TEST
7374
7375

```



```

7417
7418 037224 012700 070000          MOV #70000,R0          ;ADDR. 70000 TO R0
7419 037230 052737 001000 177746  BIS #UCB,CCR          ;ENABLE UCB
7420 037236 010220          13$: MOV R2,(R0)+          ;WRITE HIT WITH UCB WILL INVALIDATE
7421                                     ;OR WRITE 0 TO ALL HI CACHE VALID STORE
7422 037240 022700 100000          CMP #100000,R0        ;DONE?
7423 037244 001374          BNE 13$              ;NO
7424 037246 042737 001000 177746  BIC #UCB,CCR          ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
7425
7426 037254 012700 070000          MOV #70000,R0          ;ADDRESS 70000 TO R0
7427 037260          7$:
7428 037260 005710          TST (R0)              ;READ MISS TO CACHE LOCATION SPECIFIED
7429                                     ;BY R0. CLOCK VALID STORE
7430                                     ;BIT INTO CMR<12>. SHOULD BE 0.
7431                                     ;ALSO CAUSES UPDATE TO CACHE.
7432                                     ;VALID STORE LOCATION WILL BE WRITTEN
7433                                     ;WITH A 1.
7434 037262 013705 177750          MOV CMR,R5            ;SAVE CMR CONTENTS
7435 037266 005710          TST (R0)              ;READ HIT. CLOCK VALID STORE BIT TO CMR<12>
7436                                     ;SHOULD BE 1.
7437 037270 013703 177750          MOV CMR,R3            ;SAVE CMR CONTENTS
7438
7438 037274 000240          25$: NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
7438 037276 000240          NOP                  ;FOR LOOP ON ERROR

7439
7440 037300 042705 167777          BIC #167777,R5        ;INTERESTED IN BIT 12
7441 037304 005705          TST R5                ;BIT 12 SHOULD BE 0
7442 037306 001416          BEQ 8$                ;PASS
7443
7444 037310 012737 001015 177746  MOV #OFF,CCR          ;DISABLE CACHE
7445 037316 105037 177750          CLRB CMR              ;CLEAR MAINT. MODE
7446 037322 010037 047322          MOV R0,CA121          ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7447 037326 006237 047322          ASR CA121
7448
7448 037332 104000          ERROR                  ;ERROR
7448                                     ;-----
7448 037334 037332          .-2
7449
7449                                     ;VALID STORE MARCH PATTERN TEST- SET B
7450                                     ;READING CACHE VALID STORE DATA
7451                                     ;THRU CMR<12> DID NOT READ 0.
7452                                     ;THIS SUGGESTS THAT A RAM LOCATION
7453                                     ;SPECIFIED BY CA121 WAS OVERWRITTEN
7454                                     ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
7455                                     ;LOCATION.
7456                                     ;THIS INDICATES THAT VALID STORE RAM
7457                                     ;SET B IS BAD.
7458
7459 037336 047322          CA121                  ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7460 037340 000000          0
7461 037342 000430          BR 3$
7462 037344 042703 167777          8$: BIC #167777,R3        ;END THE TEST
7463 037350 022703 010000          CMP #10000,R3         ;INTERESTED IN BIT 12 ONLY
7464 037354 001416          BEQ 9$                ;BIT 12 SHOULD BE 1
7465                                     ;PASS
7466 037356 012737 001015 177746  MOV #OFF,CCR          ;DISABLE CACHE
7467 037364 105037 177750          CLRB CMR              ;CLEAR MAINT. MODE

```

```

7468 037370 010037 047322      MOV R0,CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7469 037374 006237 047322      ASR CA121
7470                                ERROR                ;ERROR
                                037400 104000              ;-----
                                037402 037400              .-2
7471                                ;VALID STORE MARCH PATTERN TEST- SET B
7472                                ;READING CACHE VALID STORE DATA
7473                                ;THRU CMR<12> DID NOT READ 1.
7474                                ;THIS SUGGESTS THAT VALID STORE RAM
7475                                ;IC SET B IS BAD.
7476                                ;THRU CMR<12>
7477                                ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
7478 037404 047322      CA121
7479 037406 000000      0
7480 037410 000405      BR 3$
7481 037412 062700 000002      9$: ADD #2,R0      ;END TEST
7482 037416 022700 100000      3$: CMP #100000,R0 ;NEXT LOCATION
7483 037422 001316      BNE 7$      ;HAS ALL HI CACHE BEEN DONE?
7484 037424 012737 001015 177746 3$: MOV #OFF,CCR ;NO,CONTINUE
7485 037432 105037 177750      CLR B CMR      ;DISABLE CACHE
7486 037436 000240      10$: NOP      ;DISABLE MAINT. MODE
                                ;END OF TEST

```

7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507

.SBTTL *
.SBTTL
.SBTTL * CACHE HIT TESTS
.SBTTL
.SBTTL
.SBTTL *

```

7508 037440 005237 001464      T204:            INC $TESTN                   ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 204
                                .SBTTL
                                .SBTTL
                                .SBTTL     VERIFY THAT THE CACHE HIT NAND GATE CAN INDICATE A READ
                                .SBTTL            HIT CONDITION.
                                .SBTTL
                                .SBTTL            PROCEDURE:        CREATE A READ HIT CONDITION TO LO CACHE
                                .SBTTL                                WITH LO CACHE ENABLED,AND VERIFY THAT
                                .SBTTL                                OUTPUT OF THE CACHE HI NAND GATE    READS 0
                                .SBTTL                                THRU CMR<8>.

```

7509
7510
7511
7512
7513
7514
7515
7516
7517

7518
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7530
7531

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

7532 037444 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

037446 037456
037450 070004
037452 070030
037454 070040
037456 012737 001015 177746
037464 004437 002352
037470 037562

40\$
1\$-40\$+70000-14
20\$-40\$+70000-14
25\$-40\$+70000-14
40\$: MOV #OFF,CCR
JSR R4,RELCTH
10\$+2

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

7533 037472 005037 060000
7534 037476 112737 000002 177750
7535
7536
7537
7538 037504 012737 000011 177746
7539
7540 037512 005737 040000
7541 037516 005737 060000
7542 037522 005737 060000
7543
7544
7545 037526 013701 177750
7546

1\$: CLR 60000
MOV B #HODO,CMR

MOV #11,CCR

TST 40000
TST 60000
20\$: TST 60000

MOV CMR,R1

25\$: NOP
NOP

;ALL 0'S TO MAIN MEMORY LOC. 60000
;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AN
;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
;ENABLE LOW CACHE FOR A READ HIT
;
;READ UPDATE TO LOW CACHE LOACATION 0000
;READ HIT; ALL INPUTS OF CACHE HIT NAND
;GATE ARE 1; CLOCK STATUS OF NAND GATE
;OUTPUT TO CMR<8>
;SAVE CMR CUNTENTS
;INSTRUCTION 'JMP 1\$' PLACED HERE
;FOR LOOP ON ERROR

7547 037536 012737 001015 177746
7548 037544 032701 000400
7549 037550 001403
7550
7551

MOV #OFF,CCR
BIT #HIT,R1
BEQ 10\$

;DISABLE CACHE
;WAS CACHE HIT SIGNAL A 0
;PASS

037552 104000
037554 037552

ERROR
.-2

;ERROR
;-----

7552
7553
7554
7555 037556 000000
7556

0

;CACHE HIT TESTS
;READING OUTPUT OF CACHE HIT NAND GATE
;THRU CMR<8> DID NOT RESULT IN A 0

:::

7557 037560 000240
 7558
 7559
 7560
 7561
 7562 037562 005237 001464

10\$: NOP ;END OF TEST

*205: INC \$TESTN ;UPDATE TEST ID
 :*****

7563
 7564
 7565
 7566
 7567
 7568
 7569
 7570
 7571
 7572
 7573
 7574
 7575
 7576
 7577
 7578
 7579
 7580
 7581
 7582
 7583
 7584

.SBTTL
 .SBTTL TEST 205
 .SBTTL
 .SBTTL
 .SBTTL CHECK FORCE MISS LOGIC.
 .SBTTL

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

7585 037566 000004

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 40\$;ERROR/LOOP ON TEST
 1\$-40\$+60000-14 ;TEST START LOCATION
 20\$-40\$+60000-14 ;LOOP ON ERROR START LOCATION
 25\$-40\$+60000-14 ;SCOPE SYNC LOCATION
 40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
 JSR R4,RELCTL ;DISABLE CACHE
 10\$+2 ;LOCATE TEST CODE TO LOW CACHE SPACE
 ;ADDRESS OF START OF NEXT TEST

037570 037600
 037572 060004
 037574 060030
 037576 060040
 037600 012737 001015 177746
 037606 004437 002324
 037612 037704

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

7586 037614 005037 070000
 7587 037620 112737 000002 177750
 7588
 7589
 7590
 7591 037626 012737 000005 177746
 7592
 7593
 7594 037634 005737 050000
 7595 037640 005737 070000
 7596 037644 005737 070000
 7597

1\$: CLR 70000 ;ALL 0'S TO MAIN MEMORY LOC. 70000
 MOV# #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AN
 ;CLOCKING OF OUTPUT OF CACHE HIT NAND
 ;GATE INTO CMR ONLY DURING THE DESTINATION
 ;ACCESS OF AN INSTRUCTION.
 MOV #5,CCR ;NO UCB SO AS TO WRITE CACHE STORES
 ;ENABLE HI CACHE FOR A READ HIT
 ;DISABLE LO CACHE
 20\$: TST 50000
 TST 70000 ;READ UPDATE TO HI CACHE LOCATION 4000
 TST 70000 ;READ HIT; ALL INPUTS OF CACHE HIT NAND
 ;GATE ARE 1; CLOCK STATUS OF NAND GATE

```

7598
7599 037650 013701 177750          MOV CMR,R1          ;OUTPUT TO CMR<8>
7600                                ;SAVE CMR CONTENTS
                                25$: NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP          ;FOR LOOP ON ERROR
7601 037660 012737 001015 177746  MOV #OFF,CCR        ;DISABLE CACHE
7602 037666 032701 000400          BIT #HIT,R          ;WAS CACHE HIT SIGNAL A 0
7603 037672 001403                    BEQ 10$            ;PASS
7604
7605 037674 104000                    ERROR              ;ERROR
                                ;-----
                                037676 037674          .-2
                                ;CACHE HIT TESTS
                                ;READING OUTPUT OF CACHE HIT NAND GATE
                                ;THRU CMR<8> DONT NOT RESULT IN A 0
7606
7607
7608
7609 037700 000000                    0
7610
7611 037702 000240                    10$: NOP          ;END OF TEST
7612
7613
7614
7615
7616 037704 005237 001464          T206: INC $TESTN    ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 206
                                .SBTTL
                                .SBTTL
                                .SBTTL VERIFY THAT 'FORCE MISS LO' WILL INHIBIT CACHE HIT NAND GATE
                                .SBTTL FROM INDICATING A CACHE HIT.
                                .SBTTL
                                ;PROCEDURE: WITH 'FORCE MISS LO' ENABLED ATTEMPT A READ HIT TO LOW CACHE
                                ;VERIFY THAT THE OUTPUT OF CACHE HIT NAND GATE
                                ;WILL READ AS A 1 THRU CMR<8>.
                                ;CONDITIONS: INPUTS CACHE HIT NAND GATE:
                                ;COMPARE 1 =1
                                ;COMPARE 2 =1
                                ;COMPARE 3 =1
                                ;VALID =1
                                ;TAG PAR. ERR =1
                                ;HI BYTE PE =1
                                ;LO BYTE PE =1
                                ;MISS HI =1
                                ;MISS LO =0
                                ;BYPASS/WRITE =1
                                ;FAULT =-1
                                ;*****
7617
7618
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639 037710 000004                    SPCOND            ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                037712 037722          40$              ;TEST START LOCATION
                                037714 070004          1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
                                037716 070030          20$-40$+70000-14 ;SCOPE SYNC. LOCATION
                                037720 070040          25$-40$+70000-14 ;LOOP ON ERROR END LOCATION

```

```

037722 012737 001015 177746      40$:  MOV #OFF,CCR           ;DISABLE CACHE
037730 004437 002352              JSR R4,RELCTH          ;LOCATE TEST CODE TO HIGH CACHE SPACE
037734 040026 10$+2               ;ADDRESS OF START OF NEXT TEST

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

7640 037736 005037 060000      1$:  CLR 60000             ;ALL 0'S TO MAIN MEMORY LOC. 60000
7641 037742 112737 000000      MOVB #HODO,CMR        ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AN
7642                                     ;CLOCKING OF OUTPUT OF CACHE HIT NAND
7643                                     ;GATE INTO CMR ONLY DURING THE DESTINATION
7644                                     ;ACCESS OF AN INSTRUCTION.
7645 037750 012737 000015 177746      MOV #15,CCR          ;NO UCB SO AS TO WRITE CACHE STORES
7646                                     ;DISABLE LOW CACHE
7647 037756 005737 040000      TST 40000
7648 037762 005737 060000      TST 60000
7649 037766 005737 060000      20$:  TST 60000        ;READ UPDATE TO LOW CACHE LOCATION 0000
7650                                     ;READ HIT;
7651                                     ;CLOCK STATUS OF NAND GATE
7652 037772 013701 177750      MOV CMR,R1          ;OUTPUT TO CMR<8>
7653                                     ;SAVE CMR CONTENTS
037776 000240      25$:  NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
040000 000240      NOP                ;FOR LOOP ON ERROR

7654 040002 012737 001015 177746      MOV #OFF,CCR        ;DISABLE CACHE
7655 040010 032701 000400      BIT #HIT,R1         ;WAS CACHE HIT SIGNAL A 1
7656 040014 001003      BNE 10$            ;PASS
7657
7658      040016 104000      ERROR              ;ERROR
;-----
040020 040016      .-2
;CACHE HIT TESTS
;READING OUTPUT OF CACHE HIT NAND GATE
;THRU CMR<8> DID NOT RESULT IN A 1

7659
7660
7661
7662 040022 000000      0
7663
7664 040024 000240      10$:  NOP              ;END OF TEST
7665
7666
7667
7668
7669
7670
7671
7672 040026 005237 001464      T207:  INC $TESTN      ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 207
.SBTTL
.SBTTL
.SBTTL VERIFY THAT 'TAG PARITY ERROR' WILL INHIBIT CACHE HIT NAND GATE
.SBTTL FROM INDICATING A CACHE HIT.
.SBTTL
;:::
CONDITIONS:  INPUTS CACHE HIT NAND GATE:
;:::
COMPARE 1    1
COMPARE 2    -1
;:::
7673
7674
7675
7676
7677
7678
7679

```

7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

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

7691 040032 000004

040034 040044
040036 070000
040040 070054
040042 070072
040044 012737 001015 177746
040052 004437 00332
040056 040212

SCPCND
40\$
1\$-40\$+70000-14
20\$-40\$+70000-14
25\$-40\$+70000-14
40\$: MOV #OFF,CCR
JSR R4,RELCTH
10\$+2

.....
: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

7692 040060 005037 000000
7693 040064 112737 000002 177750
7694
7695
7696 040072 012737 000015 177746
7697 040100 005737 000000
7698 040104 005737 040000
7699 040110 052737 002000 177746
7700 040116 005737 000000
7701
7702

1\$: CLR 0
MOVB #HODO,LMR

MOV #15,CCR
*ST 0
TST 40000
BIS #WPT,CCR
TST 0

:0'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 STOR
: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 STOR
:U S INTO DATA PARITY STORES,AND A 1

```

7704
7705 040122 042737 002004 177746      BIC #WWPT,FMLD,CCR      ;INTO TAG PARITY STORE.
7706 040130 005037 177746      CLR CME                  ;DISABLE WWPT;ENABLE LOW CACHE
7707 040134 005737 000000      20$: TST 0                ;CLEAR CME
7708
7709
7710
7711
7712
7713 040140 013701 177750      MOV CMR,R1              ;READ HIT; ALL 0'S WILL BE PLACED ON INPUTS
7714 040144 012737 001015 177746      MOV #OFF,CCR            ;OF DATA PARITY ERROR CHECK PARITY GEN'S, BUT
7715
7716 040152 000240      25$: NOP                ;TAG PARITY CHECK GENERATOR WILL
7717 040154 000240      NOP                      ;SEE ODD DATA DUE TO WRONG PARITY
7718
7719
7720
7721 040156 052737 000400 177746      BIS #FC,CCR             ;FROM PREVIOUS READ UPDATE.
7722
7723
7724 040164 032737 010000 177746      BIT #VCIP,CCR           ;OUTPUT TO CMR<8>
7725
7726 040172 001374      BNE #-6                 ;SAVE CMR CONTENTS
7727 040174 032701 000400      BIT #HIT,R1            ;DISABLE CACHE
7728
7729 040200 001003      BNE 10$                ;INSTRUCTION 'JMP 1$' PLACED HERE
7730
7731
7732
7733
7734
7735
7736 040202 104000      ERROR                   ;FOR LOOP ON ERROR
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751 040204 040202      -.2                     ;BEFORE LEAVING TEST ELIMINATE EFFECTS
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000
8001
8002
8003
8004
8005
8006
8007
8008
8009
8010
8011
8012
8013
8014
8015
8016
8017
8018
8019
8020
8021
8022
8023
8024
8025
8026
8027
8028
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038
8039
8040
8041
8042
8043
8044
8045
8046
8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083
8084
8085
8086
8087
8088
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8099
8100
8101
8102
8103
8104
8105
8106
8107
8108
8109
8110
8111
8112
8113
8114
8115
8116
8117
8118
8119
8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149
8150
8151
8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8170
8171
8172
8173
8174
8175
8176
8177
8178
8179
8180
8181
8182
8183
8184
8185
8186
8187
8188
8189
8190
8191
8192
8193
8194
8195
8196
8197
8198
8199
8200
8201
8202
8203
8204
8205
8206
8207
8208
8209
8210
8211
8212
8213
8214
8215
8216
8217
8218
8219
8220
8221
8222
8223
8224
8225
8226
8227
8228
8229
8230
8231
8232
8233
8234
8235
8236
8237
8238
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249
8250
8251
8252
8253
8254
8255
8256
8257
8258
8259
8260
8261
8262
8263
8264
8265
8266
8267
8268
8269
8270
8271
8272
8273
8274
8275
8276
8277
8278
8279
8280
8281
8282
8283
8284
8285
8286
8287
8288
8289
8290
8291
8292
8293
8294
8295
8296
8297
8298
8299
8300
8301
8302
8303
8304
8305
8306
8307
8308
8309
8310
8311
8312
8313
8314
8315
8316
8317
8318
8319
8320
8321
8322
8323
8324
8325
8326
8327
8328
8329
8330
8331
8332
8333
8334
8335
8336
8337
8338
8339
8340
8341
8342
8343
8344
8345
8346
8347
8348
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420
8421
8422
8423
8424
8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438
8439
8440
8441
8442
8443
8444
8445
8446
8447
8448
8449
8450
8451
8452
8453
8454
8455
8456
8457
8458
8459
8460
8461
8462
8463
8464
8465
8466
8467
8468
8469
8470
8471
8472
8473
8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484
8485
8486
8487
8488
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547
8548
8549
8550
8551
8552
8553
8554
8555
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619
8620
8621
8622
8623
8624
8625
8626
8627
8628
8629
8630
8631
8632
8633
8634
8635
8636
8637
8638
8639
8640
8641
8642
8643
8644
8645
8646
8647
8648
8649
8650
8651
8652
8653
8654
8655
8656
8657
8658
8659
8660
8661
8662
8663
8664
8665
8666
8667
8668
8669
8670
8671
8672
8673
8674
8675
8676
8677
8678
8679
8680
8681
8682
8683
8684
8685
8686
8687
8688
8689
8690
8691
8692
8693
8694
8695
8696
8697
8698
8699
8700
8701
8702
8703
8704
8705
8706
8707
8708
8709
8710
8711
8712
8713
8714
8715
8716
8717
8718
8719
8720
8721
8722
8723
8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734
8735
8736
8737
8738
8739
8740
8741
8742
8743
8744
8745
8746
8747
8748
8749
8750
8751
8752
8753
8754
8755
8756
8757
8758
8759
8760
8761
8762
8763
8764
8765
8766
8767
8768
8769
8770
8771
8772
8773
8774
8775
8776
8777
8778
8779
8780
8781
8782
8783
8784
8785
8786
8787
8788
8789
8790
8791
8792
8793
8794
8795
8796
8797
8798
8799
8800
8801
8802
8803
8804
8805
8806
8807
8808
8809
8810
8811
8812
8813
8814
8815
8816
8817
8818
8819
8820
8821
8822
8823
8824
8825
8826
8827
8828
8829
8830
8831
8832
8833
8834
8835
8836
8837
8838
8839
8840
8841
8842
8843
8844
8845
8846
8847
8848
8849
8850
8851
8852
8853
8854
8855
8856
8857
8858
8859
8860
8861
8862
8863
8864
8865
8866
8867
8868
8869
8870
8871
8872
8873
8874
8875
8876
8877
8878
8879
8880
8881
8882
8883
8884
8885
8886
8887
8888
8889
8890
8891
8892
8893
8894
8895
8896
8897
8898
8899
8900
8901
8902
8903
8904
8905
8906
8907
8908
8909
8910
8911
8912
8913
8914
8915
8916
8917
8918
8919
8920
8921
8922
8923
8924
8925
8926
8927
8928
8929
8930
8931
8932
8933
8934
8935
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945
8946
8947
8948
8949
8950
8951
8952
8953
8954
8955
8956
8957
8958
8959
8960
8961
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983
8984
8985
8986
8987
8988
8989
8990
8991
8992
8993
8994
8995
8996
8997
8998
8999
9000
9001
9002
9003
9004
9005
9006
9007
9008
9009
9010
9011
9012
9013
9014
9015
9016
9017
9018
9019
9020
9021
9022
9023
9024
9025
9026
9027
9028
9029
9030
9031
9032
9033
9034
9035
9036
9037
9038
9039
9040
9041
9042
9043
9044
9045
9046
9047
9048
9049
9050
9051
9052
9053
9054
9055
9056
9057
9058
9059
9060
9061
9062
9063
9064
9065
9066
9067
9068
9069
9070
9071
9072
9073
9074
9075
9076
9077
9078
9079
9080
9081
9082
9083
9084
9085
9086
9087
9088
9089
9090
9091
9092
9093
9094
9095
9096
9097
9098
9099
9100
9101
9102
9103
9104
9105
9106
9107
9108
9109
9110
9111
9112
9113
9114
9115
9116
9117
9118
9119
9120
9121
9122
9123
9124
9125
9126
9127
9128
9129
9130
9131
9132
9133
9134
9135
9136
9137
9138
9139
9140
9141
9142
9143
9144
9145
9146
9147
9148
9149
9150
9151
9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162
9163
9164
9165
9166
9167
9168
9169
9170
9171
9172
9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186
9187
9188
9189
9190
9191
9192
9193
9194
9195
9196
9197
9198
9199
9200
9201
9202
9203
9204
9205
9206
9207
9208
9209
9210
9211
9212
9213
9214
9215
9216
9217
9218
9219
9220
9221
9222
9223
9224
9225
9226
9227
9228
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238
9239
9240
9241
9242
9243
9244
9245
9246
9247
9248
9249
9250
9251
9252
9253
9254
9255
9256
9257
9258
9259
9260
9261
9262
9263
9264
9265
9266
9267
9268
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283
9284
9285
9286
9287
9288
9289
9290
9291
9292
9293
9294
9295
9296
9297
9298
9299
9300
9301
9302
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317
9318
9319
9320
9321
9322
9323
9324
9325
9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385
9386
9387
9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404
9405
9406
9407
9408
9409
9410
9411
9412
9413
9414
9415
9416
9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430
9431
9432
9433
9434
9435
9436
9437
9438
9439
9440
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471
9472
9473
9474
9475
9476
9477
9478
9479
9480
9481
9482
9483
9484
9485
9486
9487
9488
9489
9490
9491
9492
9493
9494
9495
9496
9497
9498
9499
9500
9501
9502
9503
9504
9505
9506
9507
9508
9509
9510
9511
9512
9513
9514
9515
9516
9517
9518
9519
9520
9521
9522
9523
9524
9525
9526
9527
9528
9529
9530
9531
9532
9533
9534
9535
9536
9537
9538
9539
9540
9541
9542
9543
9544
9545
9546
9547
9548
9549
9550
9551
9552
9553
9554
9555
9556
9557
9558
9559
9560
9561
9562
9563
9564
9565
9566
9567
9568
9569
9570
9571
9572
9573
9574
9575
9576
9577
9578
9579
9580
9581
9582
9583
9584
9585
9586
9587
9588
9589
9590
9591
9592
9593
9594
9595
9596
9597
9598
9599
9600
9601
9602
9603
9604
9605
9606
9607
9608
9609
9610
9611
9612
9613
9614
9615
9616
9617
9618
9619
9620
9621
9622
9623
9624
9625
9626
9627
9628
9629
9630
9631
9632
9633
9634
9635
9636
9637
9638
9639
9640
9641
9642
9643
9644
9645
9646
9647
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668
9669
9670
9671
9672
9673
9674
9675
9676
9677
9678
9679
9680
9681
9682
9683
9684
9685
9686
9687
9688
9689
9690
9691
9692
9693
9694
9695
9696
9697
9698
9699
9700
9701
9702
9703
9704
9705
9706
9707
9708
9709
9710
9711
9712
9713
9714
9715
9716
9717
9718
9719
9720
9721
9722
9723
9724
9725
9726
9727
9728
9729
9730
9731
9732
9733
9734
9735
9736
9737
9738
9739
9740
9741
9742
9743
9744
9745
9746
9747
9748
9749
9750
9751
9752
9753
9754
9755
9756
9757
9758
9759
9760
9761
9762
9763
9764
9765
9766
9767
9768
9769
9770
9771
9772
9773
9774
9775
9776
9777
9778
9779
9780
9781
9782
9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794
9795
9796
9797
9798
9799
9800
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811
9812
9813
9814
9815
9816
9817
9818
9819
9820
9821
9822
9823
9824
9825
9826
9827
9828
9829
9830
9831
9832
9833
9834
9835
9836
9837
9838
9839
9840
9841
9842
9843
9844
9845
9846
9847
9848
9849
9850
9851
9852
9853
9854
9855
9856
9857
9858
9859
9860
9861
9862
9863
9864
9865
9866
9867
9868
9869
9870
9871
9872
9873
9874
9875
9876
9877
9878
9879
9880
9881
9882
9883
9884
9885
9886
9887
9888
9889
9890
9891
9892
9893
9894
9895
9896
9897
9898
9899
9900
9901
9902
9903
9904
9905
9906
9907
9908
9909
9910
9911
9912
9913
9914
9915
9916
9917
9918
9919
9920
9921
9922
9923
9924
9925
9926
9927
9928
9929
9930
9931
9932
9933
9934
9935
9936
9937
9938
9939
9940
9941
9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961
9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999

```

```

7736 040212 005237 001464      T210: INC $TESTN          ;UPDATE TEST !D
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7
```


7751
7752
7753
7754
7755
7756
040220
040222
040224
040226
040230
040236
040242

7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783

7784
7785
7786
7787
7788
7789
7790
7791

040216 000004
040220 040230
040222 070000
040224 070054
040226 070072
040230 012737 001015 177746
040236 004437 002352
040242 040376

040244 005037 000000
040250 112737 000002 177750
040256 012737 000015 177746
040264 005737 000000
040270 005737 040000
040274 052737 000100 177746
040302 005737 000000
040306 042737 000104 177746
040314 005037 177744
040320 005737 000000
040324 013701 177750
040330 012737 001015 177746
040336 000240
040340 000240
040342 052737 000400 177746
040350 032737 010000 177746
040356 001374
040360 032701 000400
040364 001003
040366 10400C

.....

SCPCND
40\$
1\$-40\$+70000-14
20\$-40\$+70000-14
25\$-40\$+70000-14
40\$: MOV #OFF,CCR
JSR R4,RELCTH
10\$+2

1\$: CLR 0
MOV B #HODO,CMR
MOV #15,CCR
TST 0
TST 4000
BIS #WMPD,CCR
TST 0

20\$: BIC #WMPD+FMLO,CCR
CLR CME
TST 0

25\$: NOP
NOP
BIS #FC,CCR
BIT #VCIP,CCR
BNE .-6
BIT #HIT,R1
BNE 10\$
ERROR

MISS LO 1
BYPASS/WRITE 1
FAULT 1

: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 RELATED TO HI CACHE SPACE

:0'S TO MAIN MEMORY LOCATION C.
: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 STORE
:ALLOW WRITE WRONG PARITY DATA TO :0
:& 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
:INTO TAG PARITY STORE.
:DISABLE WMPD;ENABLE LO CACHE
:CLEAR CME
:READ HIT; ALL 0'S(EVEN DATA) WILL BE
:PLACED ON INPUTS
:OF TAG PARITY ERROR CHECK PARITY GEN'S, BUT
:LO & HI BYTE PARITY CHECK GENERATORS WILL
:SEE ODD DATA DUE TO WRONG PARITY
:FROM PREVIOUS READ UPDATE.
:CLOCK STATUS OF NAND GATE TO
:OUTPUT TO CMR<8>
:SAVE CMR CONTENTS
:DISABLE CACHE
:INSTRUCTION 'JMP 1\$' PLACED HERE
:FOR LOOP ON ERROR
:BEFORE LEAVING TEST ELIMINATE EFFECTS
:OF WMPD
:WAIT TILL DONE
:WAS CACHE HIT SIGNAL A 1
:PASS
:ERROR

7792 040370 040366
 7793
 7794
 7795 040372 000000
 7796
 7797 040374 000240
 7798
 7799
 7800
 7801
 7802
 7803
 7804 040376 005237 001464

.-2

 ;CACHE HIT TESTS
 ;READING OUTPUT OF CACHE HIT NAND GATE
 ;THRU CMR<8> DID NOT RESULT IN A 1

0

10\$: NOP

;END OF TEST

:::

T211: INC \$TESTN ;UPDATE TEST ID

.SBTTL
 .SBTTL TEST 211
 .SBTTL
 .SBTTL
 .SBTTL VERIFY THAT 'FORCE MISS HI' WILL INHIBIT CACHE HIT NAND GATE
 .SBTTL FROM INDICATING A CACHE HIT.
 .SBTTL

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

7805
 7806
 7807
 7808
 7809
 7810
 7811
 7812
 7813
 7814
 7815
 7816
 7817
 7818
 7819
 7820
 7821
 7822
 7823
 7824
 7825
 7826
 7827 040402 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 LOW CACHE SPACE
 ;ADDRESS OF START OF NEXT TEST

040404 040414
 040406 060004
 040410 060030
 040412 060040
 040414 012737 001015 177746
 040422 004437 002324
 040426 040520

40\$
 1\$-40\$+60000-14
 20\$-40\$+60000-14
 25\$-40\$+60000-14
 40\$: MOV #OFF,CCR
 JSR R4,RELCTL
 10\$+2

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

7828 040430 005037 070000
 7829 040434 112737 000002 177750
 7830

1\$: CLR 70000
 MOVB #HODO,CMR

;ALL 0'S TO MAIN MEMORY LOC. 70000
 ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AN
 ;CLOCKING OF OUTPUT OF CACHE HIT NAND

```

7831                                     :GATE INTO CMR ONLY DURING THE DESTINATION
7832                                     :ACCESS OF AN INSTRUCTION.
7833 040442 012737 000015 177746         MOV #15,CCR
7834                                     :NO UCB SO AS TO WRITE CACHE STORES
7835 040450 005737 050000                 TST 50000
7836 040454 005737 070000                 TST 70000
7837 040460 005737 070000         20$: TST 70000
7838                                     :READ UPDATE HI CACHE LOCATION 4000
7839 040464 013701 177750                 MOV CMR,R1
7840                                     :READ HIT; CLOCK STATUS OF NAND GATE
7840 040470 000240                         :OUTPUT TO CMR<8>
7840 040472 000240         25$: NOP
7841 040474 012737 001015 177746         MOV #OFF,CCP
7842 040502 032701 000400                 BIT #HIT,R1
7843 040506 001003                 BNE 10$
7844                                     :DISABLE CACHE
7845 040510 104000                         :WAS CACHE HIT SIGNAL A 1
7845 040512 040510         ERROR
7846                                     :PASS
7846                                     :ERROR
7847                                     :-----
7848                                     :CACHE HIT TESTS
7849 040514 000000         0
7850                                     :READING OUTPUT OF CACHE HIT NAND GATE
7851 040516 000240         10$: NOP
7852                                     :THRU CMR<8> DID NOT RESULT IN A 1
7853                                     :
7854                                     :
7855                                     :
7856                                     :
7857                                     :
7858                                     :
7859                                     :
7860 040520 005237 001464         12$: INC $TESTN
7861                                     :UPDATE TEST ID

```

```

*****

```

```

.SBTTL
.SBTTL TEST 212
.SBTTL
.SBTTL
.SBTTL
.SBTTL

```

```

.SBTTL VERIFY THAT AN 'UNCONDITIONAL CACHE BYPASS' WILL INHIBIT CACHE HIT NAND GATE
.SBTTL FROM INDICATING A CACHE HIT.

```

```

:
: PROCEDURE: CAUSE A READ HIT TO LO CACHE WITH UCB ENABLED
: 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

```

```

7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877

```


7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940

.SBTTL
.SBTTL
.SBTTL
.SBTTL
.SBTTL

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

7941 040654 000004
040656 040666
040660 070012
040662 070054
040664 070064
040666 012737 001015 177746
040674 004437 002352
040700 041016

SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$:ERROR/LOOP ON TEST
1\$-40\$+70000-14 :TEST START LOCATION
20\$-40\$+70000-14 :LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 :SCOPE SYNC. LOCATION
40\$: MOV #OFF,CCR :LOOP ON ERROR END LOCATION
JSR R4,RELCTH :DISABLE CACHE
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

7942 040702 005000
7943 040704 005037 060000
7944 040710 012701 060000
7945 040714 112737 000002 177750
7946
7947
7948
7949 040722 012737 000011 177746
7950
7951 040730 005737 040000
7952 040734 005737 060000
7953
7954 040740 052737 001000 177746
7955
7956 040746 010011
7957

CLR R0 :CLEAR R0
CLR 60000 :ALL 0'S TO MAIN MEMORY LOC. 60000
MOV #60000,R1 :ADDRESS 60000 TO R1
1\$: MOV #HODO,CMR :HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AN
 :CLOCKING OF OUTPUT OF CACHE HIT NAND
 :GATE INTO CMR ONLY DURING THE DESTINATION
 :ACCESS OF AN INSTRUCTION.
MOV #11,CCR :NO UCB SO AS TO WRITE CACHE STORES
 :ENABLE LO CACHE
TST 40000
TST 60000 :READ UPDATE; ASSURE CORRECT PARITY IS WRITTEN
 :FOR CACHE LOCATION 0000
BIS #UCB,CCR :SET UCB SO AS TO INVALIDATE CACHE LOCATIONS
 :DURING WRITE HIT
MOV R0,(R1) :CAUSE WRITE HIT TO LOCATION 60000;
 :UCB CAUSES CACHE LOC. 0000 TO BE INVALIDATED

```

7958 040750 042737 001000 177746      BIC #UCB,CCR      ;CLEAR UCB
7959
7960 040756 005737 060000      20$:  TST 60000   ;READ UPDATE CAUSED BY VALID STORE
7961                                     ;INVALIDATED; ALL INPUTS TO CACHE HIT NAND GATE
7962                                     ;ARE 1 EXCEPT VALID.CLOCK STATUS OF NAND GATE
7963                                     ;OUTPUT TO CMR<8>
7964 040762 013702 177750      MOV CMR,R2        ;SAVE CMR CONTENTS
7965
      040766 000240      25$:  NOP         ;INSTRUCTION 'JMP 1$' PLACED HERE
      040770 000240      NOP         ;FOR LOOP ON ERROR

7966 040772 012737 001015 177746      MOV #OFF,CCR      ;DISABLE CACHE
7967 041000 032702 000400      BIT #HIT,R2       ;WAS CACHE HIT SIGNAL A 1
7968 041004 001003      BNE 10$          ;PASS
7969
7970      041006 104000      ERROR          ;ERROR
      041010 041006      .-2          ;-----

7971                                     ;CACHE HIT TESTS
7972                                     ;READING OUTPUT OF CACHE HIT NAND GATE
7973                                     ;THRU CMR<8> DID NOT RESULT IN A 1
7974 041012 000000      0
7975
7976 041014 000240      10$:  NOP         ;END OF TEST   :::
7977
7978
7979
7980
7981
7982
7983 041016 005237 001464      T214:  INC $TESTN ;UPDATE TEST ID

```

```

:*****
.SBTTL
.SBTTL TEST 214
.SBTTL
.SBTTL
.SBTTL VERIFY THA 'COMPARE 1' INPUT TO CACHE HIT NAND GATE CAN INHIBIT
.SBTTL NAND GATE FROM INDICATING A CACHE HIT.
.SBTTL
:
:          PROCEDURE:  CREATE A READ UPDATE TO LOW CACHE CAUSED BY ONLY
:                      BIT 18 ON CACHE ADDRESS LINE BEING DIFFERENT
:                      FROM BIT 18 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      0
:                      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

```

7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000
8001
8002
8003
8004

```

8005
8006
8007 041022 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
                                40$:
041024 041034          40$
041026 070000          1$-40$+70000-14
041030 070006          20$-40$+70000-14
041032 070006          25$-40$+70000-14
041034 012737 001015 177746 40$: MOV #OFF,CCR
041042 004437 002352      JSR R4,RELCTH
041046 041230          10$+2

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

8008 041050 012737 070100 000004 1$: MOV #3$-1$+70000,4 ;SETUP FOR POTENTIAL TRAP
8009 041056 012737 000340 000006   MOV #340,6
8010 041064 012737 010000 172350   MOV #10000,KPAR4 ;MAP PAGE 4 FOR 128K-132K ADDRESSING
8011 041072 012700 100000          MOV #100000,R0 ;LOAD VIRTUAL ADDRESS IN R0.WHEN MEMORY
8012                                     ;MANAGEMENT IS ENABLED,PAGE 4 WILL
8013                                     ;BE SELECTED AND ADDRESS 10000000
8014                                     ;WILL BE ACCESSED.
8015 041076 012737 000001 177572   MOV #1,SRO ;ENABLE MEM. MNGMENT.
8016 041104 012737 000020 172516   MOV #20,SR3 ;ENABLE 22-BIT MAPPING
8017
8018 041112 112737 000002 177750   MOVB #HODO,CMR ;HODO ALLOWS READ HITS,UPDATES, AND
8019                                     ;CLOCKING OF OUTPUT OF CACHE HIT NAND
8020                                     ;GATE INTO CMR ONLY DURING THE DESTINATION
8021                                     ;ACCESS OF AN INSTRUCTION.
8022 041120 012737 000011 177746   MOV #11,CCR ;NO UCB SO AS TO WRITE CACHE STORES
8023                                     ;ENABLE LOW CACHE
8024 041126 005737 040000          TST 40000
8025 041132 005737 000000          TST 0
8026                                     ;READ UPDATE; ASSURE ALL 0'S IN TAG STORE
8027 041136 005710          20$: TST (R0) ;LOCATION 0000,AND CORRECT PARITY IS WRITTEN.
8028                                     ;READ UPDATE TO CACHE LOCATION 0000
8029                                     ;CAUSED BY BIT 18 ON CACHE ADDRESS LINE
8030                                     ;DIFFERENT FROM TAG STORE BIT 18.
8031                                     ;CLOCK STATUS OF NAND GATE
8032 041140 000240          NOP ;OUTPUT TO CMR<8>
8033 041142 013701 177750          MOV CMR,R1 ;SAVE CMR CONTENTS
8034 041146 000403          BR 25$
8035 041150 013701 177750          3$: MOV CMR,R1 ;IF TRAP OCCURS WILL RETURN HERE
8036 041154 022626          CMP (SP)+,(SP)+ ;ADJUST STACK AND VECTORS
8037
                                25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP ;FOR LOOP ON ERROR

8038 041162 005037 177572          CLR SRO ;DISABLE MEM. MNGMENT.
8039 041166 005037 172516          CLR SR3 ;DISABLE 22 BIT MAPPING
8040 041172 012737 001015 177746   MOV #OFF,CCR ;DISABLE CACHE
8041 041200 012737 000006 000004   MOV #6,#4 ;RESTORE VECTORS
8042 041206 005037 000006          CLR #6
8043 041212 032701 000400          BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
8044 041216 001003          BNE 10$ ;PASS
8045
8046

```

```

041220 104000          ERROR          ;ERROR
                                ;-----
041222 041220          .-2
8047                                     ;CACHE HIT TESTS
8048                                     ;READING OUTPUT OF CACHE HIT NAND GATE
8049                                     ;THRU CMR<8> DID NOT RESULT IN A 1
8050 041224 000000          0
8051                                     ;:
8052 041226 000240          10$: NOP          ;END OF TEST
8053
8054
8055
8056
8057
8058 041230 005237 001464  *215:          INC $TESTN          ;UPDATE TEST ID
                                ;:*****
                                ;.SBITL
                                ;.SBITL TFST 215
                                ;.SBITL
                                ;.SBITL
8059                                     ;.SBITL
8060                                     ;.SBITL VERIFY THAT 'COMPARE 2' INPU TTO CACHE HIT NAND GATE CAN INHIBIT
8061                                     ;.SBITL NAND GATE FROM INDICATING A CACHE HIT.
8062                                     ;.SBITL
8063                                     ;:
8064                                     ;:
8065                                     ;:
8066                                     ;:
8067                                     ;:
8068                                     ;:
8069                                     ;:
8070                                     ;:
8071                                     ;:
8072                                     ;:
8073                                     ;:
8074                                     ;:
8075                                     ;:
8076                                     ;:
8077                                     ;:
8078                                     ;:
8079                                     ;:
8080                                     ;:
8081                                     ;:
8082 041234 000004          ;:*****
                                ;SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;          ;ERROR/LOOP ON TFST
                                ;40$          ;TEST START LOCATION
                                ;1$-40$+70000-14          ;LOOP ON ERROR START LOCATION
                                ;20$-40$+70000-4          ;SCOPE SYNC. LOCATION
                                ;25$-40$+70000-14          ;LOOP ON ERROR END LOCATION
                                ;40$: MOV #OFF,CCR          ;DISABLE CACHE
                                ;JSR R4,RELCTH          ;LOCATE TEST CODE TO HIGH CACHE SPACE
                                ;10$+2          ;ADDRESS OF START OF NEXT TEST

                                ;THE FOLLOWING LOCATIONS INCLUDING 10$
                                ;ARE RELOCATED TO HI CACHE SP'CE

8083 041262 112737 000002 177750  1$: MOVB #HODO,CMR          ;HODO ALLOWS READ HITS,UPDATES, AND
8084                                     ;CLOCKING OF OUTPUT OF CACHE HIT NAND

```



```

8085
8086
8087 041270 012737 000011 177746      MOV #11,CCR
8088
8089 041276 005737 040000      TST 40000
8090 041302 005737 000000      TST 0
8091
8092 041306 005737 040000      20$: TST 40000
8093
8094
8095
8096
8097 041312 013701 177750      MOV CMR,R1
8098
      041316 000240      25$: NOP
      041320 000240      NOP
8099 041322 012737 001015 177746      MOV #OFF,CCR
8100 041330 032701 000400      BIT #HIT,R1
8101 041334 001003      BNE 10$
8102
8103
      041336 104000      ERROR
      041340 041336      .-2
8104
8105
8106
8107 041342 000000      0
8108
8109 041344 000240      10$: NOP
8110
8111
8112
8113
8114
8115
8116 041346 005237 001464      T216: INC $TESTN

```

```

;GATE INTO CMR ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;NO UCB SO AS TO WRITE 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 LOCATION 0000
;CAUSED BY BIT 14 ON CACHE ADDRESS LINE
;DIFFERENT FROM TAG STORE BIT 14.
;CLOCK STATUS OF 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
;PASS
;ERROR
;-----
;CACHE HIT TESTS
;READING OUTPUT OF CACHE HIT NAND GATE
;THRU CMR<8> DID NOT RESULT IN A 1
;END OF TEST

```

```

;*****
.SBTTL
.SBTTL TEST 216
.SBTTL
.SBTTL
.SBTTL VERIFY THAT 'COMPARE 3' INPUT TO CACHE NAND GATE WILL INHIBIT NAND GATE
.SBTTL FROM INDICATING A CACH HIT.
.SBTTL
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

```

8132
8133
8134
8135
8136
8137
8138
8139
8140

.....
.....
.....
.....
.....
.....
.....
.....
.....

TAG PAR. ERR =1
HI BYTE PE =1
LO BYTE PE =1
MISS HI =1
MISS LO =1
BYPASS/WRITE =1
FAULT =1

8141 041352 000004

041354 041364
041356 070000
041360 070024
041362 070034
041364 012737 001015 177746
041372 004437 002352
041376 041464

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$;ERROR/LOOP ON TEST
1\$-40\$+70000-14 ;TEST START LOCATION
20\$-40\$+70000-14 ;LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 ;SCOPE SYNC. LOCATION
40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTH ;DISABLE CACHE
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

8142 041400 112737 000002 177750
8143
8144
8145
8146 041406 012737 000011 177746
8147
8148 041414 005737 020000
8149 041420 005737 000000
8150
8151
8152 041424 005737 020000
8153
8154
8155
8156
8157 041430 013701 177750
8158
041434 000240
041436 000240

8159 041440 012737 001015 177746
8160 041446 032701 000400
8161 041452 001003
8162
8163
041454 104000

041456 041454

8164
8165
8166
8167 041460 000000
8168
8169 041462 000240
8170

1\$: MOVB #HODO,CMR ;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

TST 20000
TST 0 ;READ UPDATE; ASSURE ALL 0'S IN TAG
;STORE LOCATION 0000 AND CORRECT PARITY
;IS WRITTEN.
20\$: TST 20000 ;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

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

MOV #OFF,CCR ;DISABLE CACHE
BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
BNE 10\$;PASS

ERROR ;ERROR
;-2

0 ;CACHE HIT TESTS
;READING OUTPUT OF CACHE HIT NAND GATE
;THRU CMR<8> DID NOT RESULT IN A 1

10\$: NOP ;END OF TEST ;::

8171
8172
8173
8174
8175
8176
8177
8178
8179
8180
8181
8182
8183
8184

.SBTTL *
.SBTTL
.SBTTL *
.SBTTL
.SBTTL *

CPU CLOCK RESTART TESTS-CACHED DATA FROM CACHE DATA STORE

041464 005237 001464

T217: INC \$TESTN ;UPDATE TEST ID
:*****

.SBTTL
.SBTTL TEST 217
.SBTTL

8185
8186
8187
8188
8189
8190

.SBTTL VERIFY THAT A CACHE READ HIT WILL RESULT IN DATA BEING READ
.SBTTL FROM CACHE DATA STORE, ASSURING THAT THE CACHE HAS ISSUED A
.SBTTL A CPU CLOCK RESTART SIGNAL. ASSURE THAT ALL 0'S CAN BE CACHED
.SBTTL OUT OF CACHE DATA STORE.
.SBTTL

8191 041470 000004

:*****
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
40\$;ERROR/LOOP ON TEST
1\$-40\$+70000-14 ;TEST START LOCATION
0 ;LOOP ON ERROR START LOCATION
25\$-40\$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
40\$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
JSR R4,RELCTH ;DISABLE CACHE
10\$+2 ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST

041472 041502
041474 070000
041476 000000
041500 070106
041502 012737 001015 177746
041510 004437 002352
041514 041746

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

8192 041516 012701 040000
8193 041522 012702 060000
8194 041526 012737 070076 000014
8195 041534 012737 000340 000016
8196 041542 005037 002060
8197 041546 005037 002062
8198 041552 012706 060002
8199 041556 005037 060000
8200
8201 041562 012737 000340 177776
8202 041570 112737 000002 177750
8203
8204
8205
8206
8207 041576 012737 000011 177746
8208
8209 041604 000257
8210 041606 005711
8211 041610 005712

1\$: MOV #40000,R1 ;ADDRESS 40000 TO R1
MOV #60000,R2 ;ADDRESS 60000 TO R2
MOV #3\$-1\$+70000,14 ;SETUP BPT TRAP VECTORS
MOV #340,16
CLR FAIL1 ;CLEAR ERROR FLAGS
CLR FAIL2
MOV #60002,SP ;STACK POINTER NOW POINTS TO ADDRESS 60002
CLR 60000 ;PRECONDITION MAIN MEMORY ADDRESS LOCATION
;60000 WITH ALL 0'S
MOV #340,PSW ;PRECONDITION PSW TO 340
MOVB #HODO,CMR ;HODO WILL ALLOW READ HITS AND UPDATES
;ONLY DURING THE DESTINATION MEMORY ACCESS
;OF AN INSTRUCTION.
;HODO DOES NOT ALLOW A CACHE UPDATE
;TO OCCUR DUE TO WRITE UPDATES.
MOV #11,CCR ;NO BYPASS TO ALLOW WRITES TO CACHE STORES.
;ENABLE LOW CACHE
CCC ;CLEAR ALL CONDITION CODES
TST (R1)
TST (R2)
;CACHE READ UPDATE. WRITE ALL 0'S FROM


```

8260                                     ;CREATING A READ HIT BY READING ADDRESS 60000
8261                                     ;CACHED BAD DATA FROM CACHE DATA STORE
8262 041740 047334                       RECDAT
8263 041742 000000                       0
8264                                     ;:
8265 041744 000240                       10$: NOP ;END OF TEST
8266
8267
8268
8269
8270 041746 005237 001464               T220: INC $TESTN ;UPDATE TEST ID
      ;:*****
      .SBTTL
      .SBTTL TEST 220
      .SBTTL
      .SBTTL VERIFY THAT A FLOATING 1 ACROSS 0'S DATA PATTERN CAN BE CACHED
      .SBTTL FROM CACHE DATA STORE.
      .SBTTL
      ;:*****
8271                                     SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
8272                                     ;ERROR/LOOP ON TEST
8273                                     40$ ;TEST START LOCATION
8274                                     1$-40$+70000-14 ;LOOP ON ERROR START LOCATION
8275 041752 000004                       C ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      041754 041764                       25$-40$+70000-14 ;LOOP ON ERROR END LOCATION
      041756 070016                       40$: MOV #OFF,CCR ;DISABLE CACHE
      041760 000000                       JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
      041762 070040                       10$+2 ;ADDRESS OF START OF NEXT TEST
      041764 012737 001015 177746
      041772 004437 002352
      041776 042114

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

8276 042000 012701 040000               MOV #40000,R1 ;ADDRESS 40000 TO R1
8277 042004 012702 060000               MOV #60000,R2 ;ADDRESS 60000 TO R2
8278 042010 012737 000001 047332       MOV #1,FLTPAT ;1ST FLOATING 1 PATTERN: 000001
8279 042016 013737 047332 060000       1$: MOV FLTPAT,60000 ;WRITE FLOATING 1 PATTERN TO MAIN MEMORY
8280                                     ;LOCATION 60000
8281 042024 012737 000011 177746       MOV #11,CCR ;NO BYPASS TO ALLOW WRITES TO CACHE STORES.
8282                                     ;ENABLE LOW CACHE
8283 042032 005711                       TST (R1)
8284 042034 005712                       TST (R2)
8285                                     ;CACHE READ UPDATE. WRITE FLOATING 1 PATTERN FROM
8286                                     ;MAIN MEMORY LOCATION TO CACHE DATA STORE
8287 042036 011200                       MOV (R2),R0 ;LOCATION 0000.
8288                                     ;WHEN THIS INSTRUCTION READS
8289                                     ;ADDRESS 60000,A CACHE READ HIT SHOULD RESULT
8290                                     ;AND A CPU CLOCK RESTART SIGNAL SHOULD BE ISSUED.
8291                                     ;THE CPU SHOULD READ FLOATING 1 PATTERN FROM CACHE D
8292                                     ;RATHER THAN MAIN MEMORY.
      042040 000240                       25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      042042 000240                       NOP ;FOR LOOP ON ERROR

8293
8294 042044 012737 001015 177746       MOV #OFF,CCR ;DISABLE CACHE
8295 042052 020037 047332               CMP R0,FLTPAT ;WAS THE CORRECT FLOATING 1 PATTERN RECEIVED
8296 042056 001412                       BEQ 9$ ;PASS
8297

```

```

8298 042060 010037 047334      MOV R0,RECDAT      ;GET DATA RECEIVED
8299 042064 013737 047332 047320  MOV FLTPAT,EXDAT6 ;GET EXPECTED DATA
8300                                ;ERROR
                                ;-----
                                ;CPU CLOCK RESTART-CACHED DATA TESTS
8301                                ;CREATING A READ HIT BY READING ADDRESS 60000
8302                                ;RESULTED IN INCORRECT FLOATING 1 PATTERN
8303                                ;BEING CACHED FROM CACHE DATA STORE
8304                                ;PRINT FLOATING 1 PATTERN EXPECTED FROM THE
8305                                ;READ HIT TO ADDRESS 60000
8306 042076 047320      EXDAT6      ;PRINT DATA RECEIVED FROM READ HIT TO ADDRESS 60000
8307                                ;NEXT FLOATING 1 PATTERN
8308 042100 047334      RECDAT      ;IF FLOATING 1 PATTERN 100000 HAS NOT BEEN
8309 042102 000000      0           ;DONE ,CONTINUE.
8310                                ;END OF TEST
8311 042104 006337 047332      :::      9$: ASL FLTPAT
8312 042110 103342      10$: BCC 1$
8313                                ;END OF TEST
8314 042112 000240      10$: NOP
8315
8316
8317
8318
8319
8320      .SBTTL *
8321      .SBTTL
8322      .SBTTL *
8323      .SBTTL;
8324      .SBTTL
8325      .SBTTL *
8326
8327
8328
8329
8330 042114 005237 001464      T221:      INC $TESTN      ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 221
                                .SBTTL
                                .SBTTL
                                .SBTTL VERIFY THAT DMA WRITE HITS WILL INVALIDATE ALL OF LOW CACHE VALID
                                .SBTTL STORE LOCATIONS.
                                .SBTTL
                                ;*****
8331                                SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
8332                                40$      ;ERROR/LOOP ON TEST
8333                                1$-40$+70000-14 ;TEST START LOCATION
8334                                0           ;LOOP ON ERROR START LOCATION
8335                                25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
8336 042120 000004      40$:      MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
                                JSR R4,RELCTH ;DISABLE CACHE
                                10$+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

8337	042146	032737	020000	177746		BIT #VSIU,CCR	;IS SET A BEING USED?
8338	042154	001407				BEQ 7\$;YES
8339	042156	052737	000400	177746		BIS #FC,CCR	;NO; FLUSH CACHE FOR SET A
8340	042164	032737	010000	177746		BIT #VCIP,CCR	;WAIT TILL FLUSH COMPLETE
8341	042172	001374				BNE .-6	
8342	042174	012705	060000		7\$:	MOV #60000,R5	;:ADDRESS 60000 INTO R5
8343	042200	012703	040000			MOV #40000,R3	;ADDRESS 40000 INTO R3
8344							
8345	042204	132737	000200	001501		BITB #APTSIZE,\$ENVM	;WILL APT SIZE?
8346	042212	001405				BEQ 3\$;NO,GO AUTOSIZE
8347	042214	032737	010000	001504		BIT #10000,\$USWR	;DOES APT SAY TO PERFORM TEST
8348	042222	001426				BEQ 11\$;APT SAYS DO NOT PERFORM TEST
8349	042224	000504				BR 1\$;APT SAYS DO TEST
8350							
8351	042226	012737	070116	000004	3\$:	MOV #5\$-40\$+67764,4	;SETUP FOR TRAP
8352	042234	012737	000340	000006		MOV #340,6	
8353	042242	005737	170006			TST BECR1	;ACCESS UNIBUS EXERCISER
8354	042246	000240				NOP	
8355	042250	012737	000006	000004		MOV #6,@#4	;RESTORE VECTORS
8356	042256	005037	000006			CLR @#6	
8357	042262	000465				BR 1\$;UNIBUS EXERCISER IS PRESENT;PROCEED
8358							;WITH TEST

```

8360 042264 022626          5$:    CMP (SP)+,(SP)+      ;TRAP RETURN;EXERCISER NOT PRESENT
8361 042266 012737 000006 000004    MOV #6,4             ;RESTORE VECTORS
8362 042274 005037 000006          CLR 6
8363 042300 005737 001466          11$:   TST $PASS           ;IS THI SFIRST PASS?
8364 042304 001156          BNE 10$             ;SKIP MESSAGE;SKIP TEST
8365 042306 023737 000042 000046    CMP @#42,@#46       ;IS THIS ACT11 QV OR AUTO ACCEPT?
8366 042314 001552          BEQ 10$             ;YES SKIP TYPEOUT
8367 042316 104401 042324          TYPE ,65$          ;;TYPE ASCIZ STRING
      042322 000402          BR 64$             ;;GET OVER THE ASCIZ
      ;;65$: .ASCIZ <CRLF><CRLF>
8368 042330 104401 042336          TYPE ,67$          ;;TYPE ASCIZ STRING
      042334 000432          BR 66$             ;;GET OVER THE ASCIZ
      ;;67$: .ASCIZ /UNIBUS EXERCISER NOT USED- DMA TESTS NOT PERFORMED/
8369 042422 104401 042430          TYPE ,69$          ;;TYPE ASCIZ STRING
      042426 000402          BR 68$             ;;GET OVER THE ASCIZ
      ;;69$: .ASCIZ <CRLF><CRLF>
8370 042434 000502          BR 10$
8371
8372 042436 012737 000015 177746    1$:    MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE VALID STORE
8373 042444 112737 000002 177750    MOVB #HODO,CMR      ;HODO ALLOWS UPDATES ONLY DURING THE
8374                                     ;DESTINATION ACCESS OF AN INSTRUCTION
8375 042452 005723          2$:    TST (R3)+          ;UPDATE ALL LOW CACHE LOCATIONS MAKING
8376 042454 005725          TST (R5)+          ;ALL VALID STORE LOCATIONS =1
8377                                     ;COMPLETE?
8378 042456 022705 070000          CMP #70000,R5       ;NO
8379 042462 001373          BNE 2$
8380
8381 042464 042737 000002 177750    BIC #HODO,CMR       ;CLEAR HODO SO VALID STORE CAN BE WRITTEN
8382                                     ;BY UNIBUS EXERCISER.
8383 042472 012705 060000          MOV #60000,R5       ;ADDRESS 60000 INTO R5
8384 042476 012737 060000 170004    MOV #60000,BEBA     ;SETUP UNIBUS EXERCISER
8385                                     ;ADRESS
8386 042504 012737 174000 170002    MOV #-4000,BECC     ;TRANSFER COUNT
8387 042512 012737 177777 170000    MOV #177777,BEDA    ;DATA FOR WRITE XFER
8388 042520 012737 000000 170016    MOV #0,BFCR2        ;SETUP CONTROL REGISTER 2
8389 042526 012737 003045 170006    MOV #3045,BECCR1    ;SETUP CONTROL REGISTER 1;START XFER
8390 042534 105737 170006          4$:    TSTB BECR1         ;WAIT FOR EXERCISER TO COMPLETE
8391 042540 100375          BPL 4$
8392 042542 052737 000002 177750    6$:    BIS #HODO,CMR     ;IMPLEMENT HODO. ALLOWS VALID STORE
8393                                     ;DATA TO BE WRITTEN TO CMR<12> ONLY
8394                                     ;DURING THE DESTINATION MEMORY ACCESS
8395                                     ;OF AN INSTRUCTION.
8396
8397 042550 005715          1$:    TST (R5)          ;READ LOW CACHE ADDRESS
8398                                     ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
8399                                     ;WRITE VALID STORE DATA INTO CMR<12>
8400                                     ;FROM VALID STORE ADDRESS LOCATION
8401                                     ;JUST READ.
8402 042552 013701 177750          MOV CMR,R1          ;SAVE CMR DATA
8403
      042556 000240          25$:   NOP              ;INSTRUCTION 'JMP 1$' PLACED HERE
      042560 000240          NOP                ;FOR LOOP ON ERROR
8404 042562 012737 001015 177746    MOV #OFF,CCR        ;DISABLE CACHE

```



```

043214 104000          ERROR          ;ERROR
                                ;-----
8497 043216 043214    .-2
8498                                     ;DMA TESTS
8499                                     ;ATTEMPTING A DMA WRITE TO CCR REGISTER
8500                                     ;DID NOT RESULT IN A SSYNC TIMEOUT
8501 043220 000000    0
8502 043222 032701 001000 9$: BIT #1000,R1 ;CCR BIT 9 SHOULD NOT HAVE BEEN WRITTEN TO
8503                                     ;BY THE UBE
8504 043226 001403    BEQ 10$         ;PASS
8505
043230 104000          ERROR          ;ERROR
                                ;-----
043232 043230    .-2
8506                                     ;DMA TESTS
8507                                     ;ATTEMPTING A DMA WRITE TO CCR
8508                                     ;WROTE A.1 TO CCR BIT 9
8509 043234 000000    0
8510 043236 000240    10$: NOP          ;END OF TEST
8511
8512                                     .SBTTL *
8513                                     .SBTTL
8514                                     .SBTTL *          HIT REGISTER TESTS
8515                                     .SBTTL
8516                                     .SBTTL
8517                                     .SBTTL *
8518
8519
8520
8521
8522 043240 005237 001464 T223:      INC $TESTN          ;UPDATE TEST ID
                                ;*****
                                .SBTTL
                                .SBTTL TEST 223
                                .SBTTL
8523                                     .SBTTL
8524                                     .SBTTL CHECK THAT ALL SIX HIT REGISTER BITS CAN READ 0 DUE TO SIX
8525                                     .SBTTL READ MISSES
8526                                     .SBTTL
8527                                     ;*****
8528 043244 000004    SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
043246 043256    40$
043250 070010    1$-40$+70000-14    ;TEST START LOCATION
043252 000000    0                    ;LOOP ON ERROR START LOCATION
043254 070046    25$-40$+70000-14    ;SCGPE SYNC. NOT IMPLEMENTED FOR THIS TEST
043256 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
043264 004437 002352    JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
043270 043404    10$+2              ;ADDRESS OF START OF NEXT TEST

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

8529 043272 012700 040000    MOV #40000,R0          ;ADDR. 40000 TO R0
8530 043276 012701 060000    MOV #60000,R1        ;ADDR 60000 TO R1
8531 043302 112737 000002 177750 1$: MOVB #HODO,CMR      ;HODO ALLOWS HIT REGISTER TO BE CLOCKED

```

```

8532                                     ;ONLY DURING THE DESTINATION ACCESS
8533                                     ;OF AN INSTRUCTION.
8534 043310 012737 000015 177746      MOV #15,CCR      ;NO UCB SO AS TO WRITE CACHE STORES
8535
8536 043316 005710                     TST (R0)
8537 043320 005711                     TST (R1)      ;READ MISS
8538 043322 005710                     TST (R0)      ;READ MISS
8539 043324 005711                     TST (R1)      ;READ MISS
8540 043326 005710                     TST (R0)      ;READ MISS
8541 043330 005711                     TST (R1)      ;READ MISS
8542 043332 005710                     TST (R0)      ;READ MISS
8543
8544 043334 013702 177752      MOV CHR,R2      ;SAVE CHR CONTENTS
8545
      043340 000240      25$: NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
      043342 000240      NOP      ;FOR LOOP ON ERROR

8546 043344 012737 001015 177746      MOV #OFF,CCR   ;DISABLE CACHE
8547 043352 105037 177750      CLRB CMR      ;DISABLE MAINTENANCE MODE
8548
8549 043356 042702 177700      BIC #177700,R2 ;PREPARE R2 FOR CHECK
8550 043362 005702                     TST R2        ;CHR<5:0> SHOULD HAVE BEEN ALL 0'S
8551 043364 001406                     BEQ 10$       ;PASS
8552
8553 043366 010237 047340      MOV R2,CHR50   ;PREPARE FOR ERROR REPORT
8554
      043372 104000      ERROR      ;ERROR
      043374 043372      .-2        ;-----

8555                                     ;CHR<5:0> DID NOT INDICATE ALL 0'S
8556                                     ;DUE TO SIX READ MISSES
8557 043376 047340      CHR50      ;PRINT CHR<5:0> RECEIVED
8558 043400 000000      0
8559
8560 043402 000240      10$: NOP      ;END OF TEST
8561
8562
8563
8564
8565
8566 043404 005237 001464      T224: INC $TESTN ;UPDATE TEST ID
      ;:*****
      .SBTTL
      .SBTTL TEST 224
      .SBTTL
      .SBTTL
      .SBTTL VERIFY THAT BIT 05 OF CACHE HIT REGISTER CAN CONTAIN A 1 DUE TO
      .SBTTL ONE READ HIT FOLLOWED BY FIVE READ MISSES
      .SBTTL
      ;:*****
8567
8568
8569
8570
8571
8572 043410 000004      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      043412 043422      40$      ;ERROR/LOOP ON TEST
      043414 070010      1$-40$+70000-14 ;TEST START LOCATION
      043416 000000      0        ;LOOP ON ERROR START LOCATION
      043420 070046      25$-40$+70000-14 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      043422 012737 001015 177746      40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
      ;DISABLE CACHE

```



```

8618 043552 005237 001464      -225:          INC $TESTN          ;UPDATE TEST ID
;*****
.SBTTL
.SBTTL TEST 225
.SBTTL
.SBTTL
.SBTTL          THIS TEST EXERCISES CACHE BY READING MEMORY LOCATIONS FROM
.SBTTL          60000 TO 77776 WITH CACHE ON. ALL 4K OF CACHE WILL HAVE BEEN
.SBTTL          EXERCISED. EACH ADDRESS FROM 60000 TO 77776 IS LOADED WITH
.SBTTL          DATA CORRESPONDING TO ITS OWN ADDRESS.
.SBTTL
;*****
8626 043556 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          043560 043570          40$          ;ERROR/LOOP ON TEST
          043562 043570          1$          ;TEST START LOCATION
          043564 000000          0          ;LOOP ON ERROR START LOCATION
          043566 043700          25$         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          043570          40$:          ;LOOP ON ERROR END LOCATION

8627
8628 043570 012700 060000      1$:          MOV #60000,R0          ;FIRST ADDRESS
8629 043574 010010          2$:          MOV R0,(R0)          ;FILL MEMORY WITH ADDRESSES
8630 043576 005720          TST (R0)+          ;
8631 043600 020027 077776      CMP R0,#77776
8632 043604 101773          BLOS 2$
8633 043606 012700 060000      MOV #60000,R0          ;FIRST ADDRESS
8634 043612 005037 177746      CLR @#177746          ;ENABLE CACHE
8635 043616 005110          3$:          COM (R0)          ;DOUBLE COMPLEMENT DATA AND
8636 043620 005110          COM (R0)          ;MAKE SURE IT IS IN THE CACHE
8637 043622 011005          MOV (R0),R5          ;CREATE READ HIT;STORE CACHED DATA
8638
8639 043624 020500          CMP R5,R0          ;IN R5
8640 043626 001420          BEQ 5$          ;CHECK RESULTS
8641 043630 012737 001015 177746  MOV #OFF,CCR          ;PASS
8642 043636 010037 047344          MOV R0,FAILAD          ;DISABLE CACHE
8643 043642 010037 047320          MOV R0,EXDAT6          ;SAVE FAILED ADDRESS
8644 043646 010537 047334          MOV R0,EXDAT6          ;GET EXPECTED DATA
8645
          043652 104000          ERROR          ;GET RECEIVED DATA
          ;ERROR
          ;-----

          043654 043652          -2
8646 043656 047344          FAILAD          ;PRINT FAILED ADDRESS
8647 043660 047320          EXDAT6          ;PRINT EXPECTED DATA
8648 043662 047334          RECDAT          ;PRINT RECEIVED DATA
8649 043664 000000          0
8650 043666 000404          BR 25$          ;
8651 043670 005720          5$:          TST (R0)+          ;NEXT ADDRESS
8652 043672 020027 077776      CMP R0,#77776          ;FINISHED?
8653 043676 101747          BLOS 3$          ;CONTINUE
8654
          043700 000240          25$:          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
          043702 000240          NOP          ;FOR LOOP IN ERROR

8655 043704 000240          10$:          NOP          ;END OF TEST
8656
8657
8658 043706 005237 001464      T226:          - TESTN          ;UPDATE TEST ID
    
```

```

:*****
.SBTTL
.SBTTL TEST 226
.SBTTL
:*****
8659
8660
8661
8662
8663
8664
8665 043712 000004
      043714 043724
      043716 043724
      043720 000000
      043722 043754
      043724
8666 043724 012700 060000
8667
8668 043730 112737 000002 177750
8669
8670
8671
8672 043736 012737 000011 177746
8673
8674
8675 043744 005710
8676
8677
8678 043746 106210
8679
8680
8681
8682
8683
8684 043750 013701 177750
8685
      043754 000240
      043756 000240
8686 043760 105037 177750
8687 043764 012737 000000 177746
8688 043772 032701 000400
8689 043776 001003
8690
      044000 104000
      044002 044000
8691
8692
8693
8694
8695 044004 000000
8696
8697 044006 000240
8698
8699 044010 005237 001464 T227:

:*****
.SBTTL TEST DESCRIPTION:
.SBTTL VERIFY THAT THE ASRB INSTRUCTION WILL CAUSE A CACHE BYPASS
.SBTTL UNDER A READ HIT CONDITION.
.SBTTL
:*****
      SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      40$ ;ERROR/LOOP ON TEST
      1$ ;TEST START LOCATION
      0 ;LOOP ON ERROR START LOCATION
      25$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      ;LOOP ON ERROR END LOCATION
      40$:
      1$: MOV #60000,R0 ;SETUP TEST LOCATION ADDRESS
      ;IN R0
      MOVB #2,@#177750 ;HOLD ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AN
      ;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
      CLRB @#177750 ;DISABLE MAINT MODE
      MOV #0,@#177746 ;TURN ON CACHE
      BIT #400,R1 ;WAS CMR BIT 8 A 1
      BNE 10$ ;PASS
      ERROR ;ERROR
      .-2 ;-----
      0 ;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$: NOP ;END OF TEST
      T227: INC $TESTN ;UPDATE TEST ID

```

:SBTTL
:SBTTL TEST 227
:SBTTL

8700	044014	000004		SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	044016	044026		40\$:ERROR/LOOP ON TEST
	044020	044026		1\$:TEST START LOCATION
	044022	000000		0		:LOOP ON ERROR START LOCATION
	044024	044030		25\$:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	044026		40\$:			:LOOP ON ERROR END LOCATION
8701	044026	000240		1\$:	NOP	
8702						
	044030	000240		25\$:	NOP	:INSTRUCTION 'JMP 1\$' PLACED HERE
	044032	000240			NOP	:FOR LOOP ON ERROR

8703
8704
8705
8706
8707

8708						*****
8709	044034	005737	001466	ENDPAS:	TST \$PASS	:IS THIS FIRST PASS?
8710	044040	001412			BEQ 1\$:YES; INHIBIT ITERATIONS
8711	044042	032777	004000 136014		BIT #BIT11,@SWR	:IS INHIBIT ITERATIONS IMPLEMENTED
8712						:THRU SWITCH REGISTER?
8713	044050	001006			BNE 1\$:YES
8714	044052	005237	044172		INC \$ICNT	:INCREMENT PASS ITERATION COUNTER
8715	044056	023737	044170 044172		CMP \$TIMES,\$ICNT	:HAVE ALL ITERATIONS BEEN COMPLETED?
8716	044064	001037			BNE \$DOAGN	:NO,REPEAT PROGRAM
8717	044066	005037	044172	1\$:	CLR \$ICNT	:CLEAR PASS ITERATION COUNTER
8718	044072	005237	001466		INC \$PASS	:INCREMENT PASS COUNT
8719	044076	042737	100000 001466		BIC #100000,\$PASS	:DON'T ALLW A NEGATIVE #
8720	044104	104401	044112	TYPE	,65\$::TYPE ASCIZ STRING
	044110	000410		BR	64\$::GET OVER THE ASCIZ

	044132			::65\$:	.ASCIZ	<CRLF>/END OF PASS # /
8721	044132	013746	001466	64\$:	MOV	\$PASS,-(SP) ::SAVE \$PASS FOR TYPEOUT
	044136	104405			TYPDS	::GO TYPE--DECIMAL ASCII WITH SIGN

8722	044140	104401	002112		TYPE	,\$ENULL
8723	044144	013700	000042		MOV	@#42,R0
8724	044150	001405			BEQ	\$DOAGN
8725	044152	000005			RESET	
8726	044154	004710		\$ENDAD:	JSR	PC,(R0)
8727	044156	000240			NOP	
8728	044160	000240			NOP	
8729	044162	000240			NOP	
8730	044164	000137	002400	\$DOAGN:	JMP	BEGIN ;START AGAIN

8731						
8732	044170	000012		\$TIMES:	.WORD	10.
8733	044172	000000		\$ICNT:	.WORD	0
8734						
8735						

8737
8738
8739
8740

: SYSMAC ROUTINES

.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
:*
```

```

044174 105737 002101 $TYPE: TSTB $TDFLG ;; IS THERE A TERMINAL?
044200 100002 BPL 1$ ;; BR IF YES
044202 000000 HALT ;; HALT HERE IF NO TERMINAL
044204 000430 BR 3$ ;; LEAVE
044206 010046 1$: MOV R0,-(SP) ;; SAVE R0
044210 017600 000002 MOV @2(SP),R0 ;; GET ADDRESS OF ASCIZ STRING
044214 122737 000001 001500 CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
044222 001011 BNE 62$ ;; NO,GO CHECK FOR APT CONSOLE
044224 132737 000100 001501 BITB #APTSPOOL,$ENVM ;; SPOOL MESSAGE TO APT
044232 001405 BEQ 62$ ;; NO,GO CHECK FOR CONSOLE
044234 010037 044244 MOV R0,61$ ;; SETUP MESSAGE ADDRESS FOR APT
044240 004737 001612 JSR PC,$ATY3 ;; SPOOL MESSAGE TO APT
044244 000000 61$: .WORD 0 ;; MESSAGE ADDRESS
044246 132737 000040 001501 62$: BITB #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED
044254 001003 BNE 60$ ;; YES,SKIP TYPE OUT
044256 112046 2$: MOV (R0)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
044260 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
044262 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
044264 012600 60$: MOV (SP)+,R0 ;; RESTORE R0
044266 062716 000002 3$: ADD #2,(SP) ;; ADJUST RETURN PC
044272 000002 RTI ;; RETURN
044274 122716 000011 4$: CMPB #HT,(SP) ;; BRANCH IF <HT>
044300 001430 BEQ 8$
044302 122716 000200 CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
044306 001006 BNE 5$
044310 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
044312 104401 TYPE ;; TYPE A CR AND LF
044314 002107 $CRLF
044316 105037 044522 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
044322 000755 BR 2$ ;; GET NEXT CHARACTER
044324 004737 044406 5$: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
044330 123726 002100 6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
044334 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
044336 013746 002076 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
;; AND THE NULL CHAR.
044342 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
044346 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
```

```

044350 004737 044406      JSR    PC,$TYPEC      ;;GO TYPE A NULL
044354 105337 044522      DECB   $CHARCNT      ;;DO NOT COUNT AS A COUNT
044360 000770              BR     7$            ;;LOOP
    
```

;HORIZONTAL TAB PROCESSOR

```

044362 112716 000040      8$:    MOVB   #' ,(SP)      ;;REPLACE TAB WITH SPACE
044366 004737 044406      9$:    JSR    PC,$TYPEC      ;;TYPE A SPACE
044372 132737 000007 044522  BITB   #7,$CHARCNT      ;;BRANCH IF NOT AT
044400 001372              BNE    9$              ;;TAB STOP
044402 005726              TST    (SP)+          ;;POP SPACE OFF STACK
044404 000724              BR     2$            ;;GET NEXT CHARACTER
044406 105777 135460      $TYPEC: TSTB   @STPS        ;;WAIT UNTIL PRINTER IS READY
044412 100375              BPL    $TYPEC
044414 116677 000002 135452  MOVB   2(SP),@STPB      ;;LOAD CHAR TO BE TYPED INTO DATA REG.
044422 105777 135440      TSTB   @STKS          ;;SEE IF KEYBOARD IS TALKING.
044426 100021              BPL    2$            ;;BRANCH IF IT ISN'T.
044430 017746 135434      MOV    @STKB,-(SP)      ;;PUSH CHARACTER ONTO STACK.
044434 042716 177600      BIC    #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044440 022726 000023      CMP    #23,(SP)+      ;;SEE IF THIS IS A ^S.
044444 001012              BNE    2$            ;;BRANCH TO CONTINUE IF IT ISN'T.
044446 105777 135414      3$:    TSTB   @STKS          ;;WAIT FOR ANOTHER INPUT.
044452 100375              BPL    3$            ;;BRANCH BACK IF NOT READY.
044454 017746 135410      MOV    @STKB,-(SP)      ;;PUSH NEXT CHARACTER ON STACK.
044460 042716 177600      BIC    #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044464 022726 000021      CMP    #21,(SP)+      ;;SEE IF THIS IS A ^Q.
044470 001366              BNE    3$            ;;BRANCH BACK FOR MORE WAIT IF NOT.
044472 122766 000015 000002 2$:    CMPB   #CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
044500 001003              BNE    1$            ;;BRANCH IF NO
044502 105037 044522      CLRB   $CHARCNT      ;;YES--CLEAR CHARACTER COUNT
044506 000406              BR     $TYPEX        ;;EXIT
044510 122766 000012 000002 1$:    CMPB   #LF,2(SP)      ;;IS CHARACTER A LINE FEED?
044516 001402              BEQ    $TYPEX        ;;BRANCH IF YES
044520 105227              INCB   (PC)+          ;;COUNT THE CHARACTER
044522 000000      $CHARCNT: .WORD 0      ;;CHARACTER COUNT STORAGE
044524 000207      $TYPEX: RTS    PC
    
```

8741

.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
;*(ALL:
;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
;*      TYPOC    ;;CALL FOR TYPEOUT

044526 017646 000000          $TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
044532 116637 000001 044751  MOVB     1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
044540 112637 044753          MOVB     (SP)+, $OMODE+1      ;;NUMBER OF DIGITS TO TYPE
044544 062716 000002          ADD      #2, (SP)           ;;ADJUST RETURN ADDRESS
044550 000406                    BR      $TYPON
044552 112737 000001 044751  $TYPOC: MOVB     #1, $OFILL      ;;SET THE ZERO FILL SWITCH
044560 112737 000006 044753  MOVB     #6, $OMODE+1      ;;SET FOR SIX(6) DIGITS
044566 112737 000005 044750  $TYPON: MOVB     #5, $OCNT      ;;SET THE ITERATION COUNT
044574 010346                    MOV      R3, -(SP)         ;;SAVE R3
044576 010446                    MOV      R4, -(SP)         ;;SAVE R4
044600 010546                    MOV      R5, -(SP)         ;;SAVE R5
044602 113704 044753          MOVB     $OMODE+1, R4      ;;GET THE NUMBER OF DIGITS TO TYPE
044606 005404                    NEG      R4
044610 062704 000006          ADD      #6, R4           ;;SUBTRACT IT FOR MAX. ALLOWED
044614 110437 044752          MOVB     R4, $OMODE      ;;SAVE IT FOR USE
044620 113704 044751          MOVB     $OFILL, R4       ;;GET THE ZERO FILL SWITCH
044624 016605 000012          MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
044630 005003                    CLR      R3              ;;CLEAR THE OUTPUT WORD
044632 006105                    1$:     ROL      R5           ;;ROTATE MSB INTO 'C'
044634 000404                    BR      3$              ;;GO DO MSB
044636 006105                    2$:     ROL      R5           ;;FORM THIS DIGIT
044640 006105                    ROL      R5
044642 006105                    ROL      R5
044644 010503                    MOV      R5, R3
044646 006103                    3$:     ROL      R3           ;;GET LSB OF THIS DIGIT
044650 105337 044752          DECB     $OMODE          ;;TYPE THIS DIGIT?
044654 100016                    BPL     7$              ;;BR IF NO
044656 042703 177770          BIC      #177770, R3     ;;GET RID OF JUNK
044662 001002                    BNE     4$              ;;TEST FOR 0
044664 005704                    TST     R4              ;;SUPPRESS THIS 0?
044666 001403                    BEQ     5$              ;;BR IF YES
044670 005204                    4$:     INC      R4           ;;DON'T SUPPRESS ANYMORE 0'S
044672 052703 000060          BIS      #'0, R3        ;;MAKE THIS DIGIT ASCII
044676 052703 000040          5$:     BIS      #' , R3      ;;MAKE ASCII IF NOT ALREADY
044702 110337 044746          MOVB     R3, 8$         ;;SAVE FOR TYPING
044706 104401 044746          TYPE     8$           ;;GO TYPE THIS DIGIT
044712 105337 044750          7$:     DECB     $OCNT      ;;COUNT BY 1
044716 003347                    BGT     2$              ;;BR IF MORE TO DO
044720 002402                    BLT     6$              ;;BR IF DONE
044722 005204                    INC      R4           ;;INSURE LAST DIGIT ISN'T A BLANK
044724 000744                    BR      2$              ;;GO DO THE LAST DIGIT
044726 012605                    6$:     MOV      (SP)+, R5     ;;RESTORE R5
044730 012604                    MOV      (SP)+, R4     ;;RESTORE R4
044732 012603                    MOV      (SP)+, R3     ;;RESTORE R3
044734 016666 000002 000004  MOV      2(SP), 4(SP)    ;;SET THE STACK FOR RETURNING
044742 012616                    MOV      (SP)+, (SP)
044744 000002                    RTI
044746 000                    8$:     .BYTE    0           ;;RETURN
044747 000                    .BYTE    0           ;;STORAGE FOR ASCII DIGIT
044750 000                    $OCNT:   .BYTE    0           ;;TERMINATOR FOR TYPE ROUTINE
044751 000                    $OFILL:  .BYTE    0           ;;OCTAL DIGIT COUNTER
044752 000000                    $OMODE:  .WORD    0           ;;ZERO FILL SWITCH
;;NUMBER OF DIGITS TO TYPE

```

8742

.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

```

```

044754          $TYPDS:
044754 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
044756 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
044760 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
044762 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
044764 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
044766 012746 020200  MOV      #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
044772 016605 000020  MOV      20(SP),R5      ;;GET THE INPUT NUMBER
044776 100004      BPL      1$                ;;BR IF INPUT IS POS.
045000 005405      NEG      R5                ;;MAKE THE BINARY NUMBER POS.
045002 112766 000055 000001  MOVVB   #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.
045010 005000      1$:      CLR      R0                ;;ZERO THE CONSTANTS INDEX
045012 012703 045170      MOV      #$DBLK,R3      ;;SETUP THE OUTPUT POINTER
045016 112723 000040      MOVVB   #' ,(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
045022 005002      2$:      CLR      R2                ;;CLEAR THE BCD NUMBER
045024 016001 045160      MOV      $DTBL(R0),R1    ;;GET THE CONSTANT
045030 160105      3$:      SUB      R1,R5      ;;FORM THIS BCD DIGIT
045032 002402      BLT      4$                ;;BR IF DONE
045034 005202      INC      R2                ;;INCREASE THE BCD DIGIT BY 1
045036 000774      BR      3$
045040 060105      4$:      ADD      R1,R5      ;;ADD BACK THE CONSTANT
045042 005702      TST      R2                ;;CHECK IF BCD DIGIT=0
045044 001002      BNE      5$                ;;FALL THROUGH IF 0
045046 105716      TSTB   (SP)              ;;STILL DOING LEADING 0'S?
045050 100407      BMI      7$                ;;BR IF YES
045052 106316      5$:      ASLB   (SP)              ;;MSD?
045054 103003      BCC      6$                ;;BR IF NO
045056 116663 000001 177777  MOVVB   1(SP),-1(R3)    ;;YES--SET THE SIGN
045064 052702 000060      6$:      BIS      #'0,R2      ;;MAKE THE BCD DIGIT ASCII
045070 052702 000040      7$:      BIS      #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
045074 110223      MOVVB   R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
045076 005720      TST      (R0)+      ;;JUST INCREMENTING
045100 020027 000010      CMP      R0,#10      ;;CHECK THE TABLE INDEX
045104 002746      BLT      2$                ;;GO DO THE NEXT DIGIT
045106 003002      BGT      8$                ;;GO TO EXIT
045110 010502      MOV      R5,R2      ;;GET THE LSD
045112 000764      BR      6$                ;;GO CHANGE TO ASCII
045114 105726      8$:      TSTB   (SP)+      ;;WAS THE LSD THE FIRST NON-ZERO?
045116 100003      BPL      9$                ;;BR IF NO
045120 116663 177777 177776  MOVVB   -1(SP),-2(R3)  ;;YES--SET THE SIGN FOR TYPING
045126 105013      9$:      CLRB   (R3)              ;;SET THE TERMINATOR
045130 012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
045132 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
045134 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
045136 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1

```

```

045140 012600          MOV (SP)+,R0          ;;POP STACK INTO R0
045142 104401 045170  TYPE $DBLK          ;;NOW TYPE THE NUMBER
045146 016666 000002 000004  MOV 2(SP),4(SP)      ;;ADJUST THE STACK
045154 012616          MOV (SP)+,(SP)
045156 000002          RTI                          ;;RETURN TO USER
045160 023420          $DTBL: 1000.
045162 001750          1000.
045164 000144          100.
045166 000012          10.
045170          $DBLK: .BLKW 4
8743          .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.
  
```

```

045200 010046          $TRAP: MOV R0,-(SP)          ;;SAVE R0
045202 016600 000002  MOV 2(SP),R0          ;;GET TRAP ADDRESS
045206 005740          TST -(R0)          ;;BACKUP BY 2
045210 111000          MOVB (R0),R0          ;;GET RIGHT BYTE OF TRAP
045212 006300          ASL R0          ;;POSITION FOR INDEXING
045214 016000 045234  MOV $TRPAD(R0),R0      ;;INDEX TO TABLE
045220 000200          RTS R0          ;;GO TO ROUTINE
  
```

```

;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
  
```

```

045222 011646          $TRAP2: MOV (SP),-(SP)          ;;MOVE THE PC DOWN
045224 016666 000004 000002  MOV 4(SP),2(SP)      ;;MOVE THE PSW DOWN
045232 000002          RTI                          ;;RESTORE THE PSW
  
```

```

.SBTTL TRAP TABLE
  
```

```

*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
*BY THE 'TRAP' INSTRUCTION.
  
```

	ROUTINE	
045234	045222	\$TRPAD: .WORD \$TRAP2
045236	044174	\$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
045240	044552	\$TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
045242	044526	\$TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
045244	044566	\$TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
045246	044754	\$TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)

```

8744
8745          ;;*****
8746
8747          ERROR=104000
8748 045250 000000  ERRPC: .WORD 0
8749
8750 045252 010037 045304  $ERROR: MOV R0,SAVR0          ;SAVE R0 THRU R5
8751 045256 010137 045306  MOV R1,SAVR1
8752 045262 010237 045310  MOV R2,SAVR2
  
```

```

8753 045266 010337 045312      MOV R3,SAVR3
8754 045272 010437 045314      MOV R4,SAVR4
8755 045276 010537 045316      MOV R5,SAVR5
8756 045302 000406              BR TITL
8757 045304 000000              SAVR0: .WORD
8758 045306 000000              SAVR1: .WORD
8759 045310 000000              SAVR2: .WORD
8760 045312 000000              SAVR3: .WORD
8761 045314 000000              SAVR4: .WORD
8762 045316 000000              SAVR5: .WORD
8763
8764
8765 045320 032777 020000 134536 TITL:      BIT #BIT13,@SWR      ;INHIBIT ERROR TYPEOUTS?
8766 045326 001113              BNE INHERR          ;YES
8767
8768 045330 011601              MOV (SP),R1         ;R1 CONTAINS ADDRESS FOLLOWING ERRPC ADDRESS
8769 045332 012137 045250      MOV (R1)+,ERRPC    ;LOAD ERRPC ADDRESS AND POINT
8770                                     ;TO NEXT ARGUMENT
8771 045336 104401 045344      TYPE ,65$          ;;TYPE ASCIZ STRING
      045342 000411              BR 64$             ;;GET OVER THE ASCIZ
      045366                      ;;65$: .ASCIZ <CRLF><CRLF>/TESTNO ERRPC/
      64$:
8772
8773 045366 005711              3$: TST (R1)        ;END OF ARGUMENTS?
8774 045370 001412              BEQ DAT            ;YES,GO PRINT DATA
8775 045372 012702 047236      MOV #PRTABL,R2     ;ADDRESS OF START OF PRINT TABLE LIST
8776 045376 012703 045742      MOV #PRTITL,R3    ;ADDRESS OF START OF TITLES
8777 045402 005723              2$: TST (R3)+      ;INDEX THRU TITLES
8778 045404 021122              CMP (R1),(R2)+    ;SEARCH PRINT TABLE LIST FOR TITLE
8779 045406 001375              BNE 2$            ;NO; CHECK NEXT LOCATION IN LIST
8780
8781 045410 004753              JSR PC,@-(R3)     ;FOUND IT; GO PRINT TITLE
8782 045412 005721              TST (R1)+        ;R1 POINTS TO NXT ARGUMENT IN TEST CODE
8783 045414 000764              BR 3$
8784
8785
8786 045416                      DAT:
      045416 104401 045424      TYPE ,65$          ;;TYPE ASCIZ STRING
      045422 000401              BR 64$             ;;GET OVER THE ASCIZ
      64$:
      045426                      ;;65$: .ASCIZ <CRLF>
8787 045426 013746 001464      MOV $TESTN,-(SP)  ;;SAVE $TESTN FOR TYPEOUT
      045432 104403              TYPOS              ;;GO TYPE--OCTAL ASCII
      045434 006                .BYTE 6            ;;TYPE 6 DIGIT(S)
      045435 000                .BYTE 0            ;;SUPPRESS LEADING ZEROS
8788 045436 104401 045444      TYPE ,67$          ;;TYPE ASCIZ STRING
      045442 000402              BR 66$             ;;GET OVER THE ASCIZ
      66$:
      045450                      ;;67$: .ASCIZ / /
8789 045450 013746 045250      MOV ERRPC,-(SP)   ;;SAVE ERRPC FOR TYPEOUT
      045454 104402              TYPOC              ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8790 045456 104401 002112      TYPE , $NULL
8791
8792 045462 011601              MOV (SP),R1       ;R1 CONTAINS ADDRESS FOLLOWING ERRORPC ADDRESS
8793 045464 005721              TST (R1)+        ;POINT TO NEXT ARGUMENT
8794 045466 005711              3$: TST (R1)      ;END OF ARGUMENTS?
8795 045470 001437              BEQ FIN           ;YES

```

8796	045472	012702	047236		MOV #PRTABL,R2	;ADDRESS OF START OF PRINT TABLE LIST
8797	045476	012703	046714		MOV #PRDATA,R3	;ADDRESS OF STERT OF PRINT DATA
8798	045502	005723		2\$:	TST (R3)+	;INDEX THRU DATA PRINTS
8799	045504	021122			CMP (R1),(R2)+	;SEARCH PRINT TABLE LIST FOR TITLE
8800	045506	001375			BNE 2\$;NO; CHECK NEXT LOCATION IN LIST
8801	045510	104401	045516		TYPE ,69\$::TYPE ASCIZ STRING
	045514	000404			BR 68\$::GET OVER THE ASCIZ
				::69\$:	.ASCIZ / /	
				68\$:		
8802	045526	004753			JSR PC,@-(R3)	:
8803	045530	104401	002112		TYPE ,\$NULL	
8804	045534	104401	045542		TYPE ,71\$::TYPE ASCIZ STRING
	045540	000404			BR 70\$::GET OVER THE ASCIZ
				::71\$:	.ASCIZ / /	
				70\$:		
8805	045552	005721			TST (R1)+	;R1 POINTS TO NEXT ARGUEMENT
8806	045554	000744			BR 3\$	
8807						
8808					;INHIBIT ERROR TYPEOUT CODE	
8809						
8810	045556	011601		INHERR:	MOV (SP),R1	;R1 CONTAINS ADDRESS FOLLOWING
8811						;ERRPC ADDRESS
8812	045560	005711		1\$:	TST (R1)	;IS THIS THE END OF ARGUEMENT LIST?
8813	045562	001402			BEQ FIN	;YES
8814	045564	005721			TST (R1)+	;POINT TO NEXT ARGUEMENT
8815	045566	000774			BR 1\$	
8816						
8817						
8818	045570	005721		FIN:	TST (R1)+	;R1 NOW CONTAINS RETURN ADDRESS
8819	045572	010116			MOV R1,(SP)	;SETUP RETURN ADDRESS IN STACK
8820						
8821					;HOE,LOE OPTION DETERMINATION	
8822						
8823	045574	122737	000001	001500	CMPB #APTENV,\$ENV	;IS THIS APT?
8824	045602	001410			BEQ 2\$;YES HALT ON ERROR
8825	045604	023737	000042	000046	CMP @#42,@#46	;IS THIS ACT11 QV OR AUTO ACCEPT
8826	045612	001404			BEQ 2\$;YES HALT ON ERROR
8827	045614	032777	100000	134242	BIT #BIT15,@SWR	;IS HALT ON ERROR IMPLEMENTED?
8828	045622	001401			BEQ 1\$;NO
8829	045624	000000			2\$:	HALT
8830	045626	032777	001000	134230	1\$:	BIT #BIT09,@SWR
8831	045634	001010			BNE 4\$;YES
8832						;NO;INITIALIZE LOCATIONS WHERE
8833						; 'JMP 1\$' IS PLACED FOR LOOP ON ERROR
8834						;WITH NOP'S
8835	045636	012777	000240	134260	MOV #240,@ADRJMP	
8836	045644	012777	000240	134254	MOV #240,@ADR1\$	
8837	045652	000137	045710		JMP 5\$;CONTINUE WITH PRESENT TEST
8838						
8839	045656	012777	000137	134240	4\$:	MOV #137,@ADRJMP
8840	045664	013777	002120	134234		;WRITE 'JMP' INSTRUCTION TO PROPER ADDRESS
8841	045672	013737	0021??	177752	MOV @#STR1LP,@ADR1\$;WRITE '1\$' LOCATION TO PROPER ADDRESS
8842	045700	105037	177751		MOV ADRSYNC,CHR	;LOAD ADDRESS LOCATION FOR SCOPE SYNC
8843	045704	013716	002120		CLRB CMR+1	
8844					MOV @#STR1LP,(SP)	;SETUP LOCATION FOR LOOP ON ERROR
8845						;IN STACK
8846						

```

8847 045710 013700 045304          5$:  MOV SAVR0,R0          ;RESTORE REGISTERS
8848 045714 013701 045306          MOV SAVR1,R1
8849 045720 013702 045310          MOV SAVR2,R2
8850 045724 013703 045312          MOV SAVR3,R3
8851 045730 013704 045314          MOV SAVR4,R4
8852 045734 013705 045316          MOV SAVR5,R5
8853
8854 045740 000002          RTI          ;RETURN
8855 045742 046004          PRTITL: 1$
8856 045744 046040          2$
8857 045746 046074          3$
8858 045750 046130          4$
8859 045752 046164          5$
8860 045754 046220          6$
8861 045756 046254          7$
8862 045760 046310          8$
8863 045762 046344          9$
8864 045764 046400          10$
8865 045766 046434          11$
8866 045770 046470          12$
8867 045772 046524          13$
8868 045774 046550          14$
8869 045776 046602          15$
8870 046000 046630          16$
8871 046002 046662          17$
8872 046004          1$:
      046004 104401 046012          TYPE ,65$          ;;TYPE ASCIZ STRING
      046010 000412          BR ,64$          ;;GET OVER THE ASCIZ
      ;;65$: .ASCIZ / CA210(21:0) /
8873 046036          64$:
      046036 000207          RTS PC
8874 046040          2$:
      046040 104401 046046          TYPE ,67$          ;;TYPE ASCIZ STRING
      046044 000412          BR ,66$          ;;GET OVER THE ASCIZ
      ;;67$: .ASCIZ / AMR210(21:0) /
8875 046072          66$:
      046072 000207          RTS PC
8876 046074          3$:
      046074 104401 046102          TYPE ,69$          ;;TYPE ASCIZ STRING
      046100 000412          BR ,68$          ;;GET OVER THE ASCIZ
      ;;69$: .ASCIZ / CHR157(15:07) /
8877 046126          68$:
      046126 000207          RTS PC
8878 046130          4$:
      046130 104401 046136          TYPE ,71$          ;;TYPE ASCIZ STRING
      046134 000412          BR ,70$          ;;GET OVER THE ASCIZ
      ;;71$: .ASCIZ / CA2113(21:13) /
8879 046162          70$:
      046162 000207          RTS PC
8880 046164          5$:
      046164 104401 046172          TYPE ,73$          ;;TYPE ASCIZ STRING
      046170 000412          BR ,72$          ;;GET OVER THE ASCIZ
      ;;73$: .ASCIZ / CHR80(8:0) /
8881 046216          72$:
      046216 000207          RTS PC
8882 046220          6$:
      046220 104401 046226          TYPE ,75$          ;;TYPE ASCIZ STRING

```


	046224	000412		BR	74\$::GET OVER THE ASCIZ
			::75\$:	.ASCIZ	/ CDR150(15:0)	/
	046252		74\$:			
8883	046252	000207			RTS PC	
8884	046254			7\$:		
	046254	104401	046262	TYPE	77\$::TYPE ASCIZ STRING
	046260	000412		BR	76\$::GET OVER THE ASCIZ
			::77\$:	.ASCIZ	/ EXDAT6	/
	046306		76\$:			
8885	046306	000207			RTS PC	
8886	046310			8\$:		
	046310	104401	046316	TYPE	79\$::TYPE ASCIZ STRING
	046314	000412		BR	78\$::GET OVER THE ASCIZ
			::79\$:	.ASCIZ	/ CA121(12:1)	/
	046342		78\$:			
8887	046342	000207			RTS PC	
8888	046344			9\$:		
	046344	104401	046352	TYPE	81\$::TYPE ASCIZ STRING
	046350	000412		BR	80\$::GET OVER THE ASCIZ
			::81\$:	.ASCIZ	/ EXDAT1	/
	046376		80\$:			
8889	046376	000207			RTS PC	
8890	046400			10\$:		
	046400	104401	046406	TYPE	83\$::TYPE ASCIZ STRING
	046404	000412		BR	82\$::GET OVER THE ASCIZ
			::83\$:	.ASCIZ	/ CM1513(15:13)	/
	046432		82\$:			
8891	046432	000207			RTS PC	
8892	046434			11\$:		
	046434	104401	046442	TYPE	85\$::TYPE ASCIZ STRING
	046440	000412		BR	84\$::GET OVER THE ASCIZ
			::85\$:	.ASCIZ	/ CNT121(12:1)	/
	046466		84\$:			
8893	046466	000207			RTS PC	
8894	046470			12\$:		
	046470	104401	046476	TYPE	87\$::TYPE ASCIZ STRING
	046474	000412		BR	86\$::GET OVER THE ASCIZ
			::87\$:	.ASCIZ	/ FLTPAT	/
	046522		86\$:			
8895	046522	000207			RTS PC	
8896	046524			13\$:		
	046524	104401	046532	TYPE	89\$::TYPE ASCIZ STRING
	046530	000406		BR	88\$::GET OVER THE ASCIZ
			::89\$:	.ASCIZ	/ RECDAT	/
	046546		88\$:			
8897	046546	000207			RTS PC	
8898	046550			14\$:		
	046550	104401	046556	TYPE	91\$::TYPE ASCIZ STRING
	046554	000411		BR	90\$::GET OVER THE ASCIZ
			::91\$:	.ASCIZ	/ EXDAT3	/
	046600		90\$:			
8899	046600	000207			RTS PC	
8900	046602			15\$:		
	046602	104401	046610	TYPE	93\$::TYPE ASCIZ STRING
	046606	000407		BR	92\$::GET OVER THE ASCIZ
			::93\$:	.ASCIZ	/ CHR50	/
	046626		92\$:			

8901	046626	000207			RTS PC	
8902	046630			16\$:		
	046630	104401	046636	TYPE	95\$::TYPE ASCIZ STRING
	046634	000411		BR	94\$::GET OVER THE ASCIZ
				::95\$:	.ASCIZ /	CMR119 /
	046660			94\$:		
8903	046660	000207			RTS PC	
8904	046662			17\$:		
	046662	104401	046670	TYPE	97\$::TYPE ASCIZ STRING
	046666	000411		BR	96\$::GET OVER THE ASCIZ
				::97\$:	.ASCIZ /	FAILAD /
	046712			96\$:		
8905	046712	000207			RTS PC	
8906					.EVEN	
8907						
8908						
8909	046714	046756		PRDATA: 1\$		
8910	046716	047000		2\$		
8911	046720	047022		3\$		
8912	046722	047034		4\$		
8913	046724	047046		5\$		
8914	046726	047060		6\$		
8915	046730	047070		7\$		
8916	046732	047100		8\$		
8917	046734	047112		9\$		
8918	046736	047124		10\$		
8919	046740	047136		11\$		
8920	046742	047150		12\$		
8921	046744	047160		13\$		
8922	046746	047170		14\$		
8923	046750	047202		15\$		
8924	046752	047214		16\$		
8925	046754	047226		17\$		
8926	046756			1\$:		
	046756	013746	047300	MOV	CA210,-(SP)	::SAVE CA210 FOR TYPEOUT
	046762	104403		TYPOS		::GO TYPE--OCTAL ASCII
	046764	003		.BYTE	3	::TYPE 3 DIGIT(S)
	046765	001		.BYTE	1	::TYPE LEADING ZEROS
8927	046766	013746	047302	MOV	CA210+2,-(SP)	::SAVE CA210+2 FOR TYPEOUT
	046772	104403		TYPOS		::GO TYPE--OCTAL ASCII
	046774	005		.BYTE	5	::TYPE 5 DIGIT(S)
	046775	001		.BYTE	1	::TYPE LEADING ZEROS
8928	046776	000207			RTS PC	
8929	047000			2\$:		
	047000	013746	047304	MOV	AMR210,-(SP)	::SAVE AMR210 FOR TYPEOUT
	047004	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047006	003		.BYTE	3	::TYPE 3 DIGIT(S)
	047007	001		.BYTE	1	::TYPE LEADING ZEROS
8930	047010	013746	047306	MOV	AMR210+2,-(SP)	::SAVE AMR210+2 FOR TYPEOUT
	047014	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047016	005		.BYTE	5	::TYPE 5 DIGIT(S)
	047017	001		.BYTE	1	::TYPE LEADING ZEROS
8931	047020	000207			RTS PC	
8932	047022			3\$:		
	047022	013746	047310	MOV	CHR157,-(SP)	::SAVE CHR157 FOR TYPEOUT
	047026	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047030	003		.BYTE	3	::TYPE 3 DIGIT(S)

8933	047031	001		.BYTE	1		::TYPE LEADING ZEROS
	047032	000207				RTS PC	
8934	047034		047312	4\$:			
	047034	013746		MOV		CA2113,-(SP)	::SAVE CA2113 FOR TYPEOUT
	047040	104403		TYPOS			::GO TYPE--OCTAL ASCII
	047042	003		.BYTE	3		::TYPE 3 DIGIT(S)
	047043	001		.BYTE	1		::TYPE LEADING ZEROS
8935	047044	000207				RTS PC	
8936	047046		047314	5\$:			
	047046	013746		MOV		CHR80,-(SP)	::SAVE CHR80 FOR TYPEOUT
	047052	104403		TYPOS			::GO TYPE--OCTAL ASCII
	047054	003		.BYTE	3		::TYPE 3 DIGIT(S)
	047055	001		.BYTE	1		::TYPE LEADING ZEROS
8937	047056	000207				RTS PC	
8938	047060		047316	6\$:			
	047060	013746		MOV		CDR150,-(SP)	::SAVE CDR150 FOR TYPEOUT
	047064	104402		TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)
8939	047066	000207				RTS PC	
8940	047070		047320	7\$:			
	047070	013746		MOV		EXDAT6,-(SP)	::SAVE EXDAT6 FOR TYPEOUT
	047074	104402		TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)
8941	047076	000207				RTS PC	
8942	047100		047322	8\$:			
	047100	013746		MOV		CA121,-(SP)	::SAVE CA121 FOR TYPEOUT
	047104	104403		TYPOS			::GO TYPE--OCTAL ASCII
	047106	004		.BYTE	4		::TYPE 4 DIGIT(S)
	047107	001		.BYTE	1		::TYPE LEADING ZEROS
8943	047110	000207				RTS PC	
8944	047112		047324	9\$:			
	047112	013746		MOV		EXDAT1,-(SP)	::SAVE EXDAT1 FOR TYPEOUT
	047116	104403		TYPOS			::GO TYPE--OCTAL ASCII
	047120	001		.BYTE	1		::TYPE 1 DIGIT(S)
	047121	001		.BYTE	1		::TYPE LEADING ZEROS
8945	047122	000207				RTS PC	
8946	047124		047326	10\$:			
	047124	013746		MOV		CM1513,-(SP)	::SAVE CM1513 FOR TYPEOUT
	047130	104403		TYPOS			::GO TYPE--OCTAL ASCII
	047132	001		.BYTE	1		::TYPE 1 DIGIT(S)
	047133	001		.BYTE	1		::TYPE LEADING ZEROS
8947	047134	000207				RTS PC	
8948	047136		047330	11\$:			
	047136	013746		MOV		CNT121,-(SP)	::SAVE CNT121 FOR TYPEOUT
	047142	104403		TYPOS			::GO TYPE--OCTAL ASCII
	047144	004		.BYTE	4		::TYPE 4 DIGIT(S)
	047145	001		.BYTE	1		::TYPE LEADING ZEROS
8949	047146	000207				RTS PC	
8950	047150		047332	12\$:			
	047150	013746		MOV		FLTPAT,-(SP)	::SAVE FLTPAT FOR TYPEOUT
	047154	104402		TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)
8951	047156	000207				RTS PC	
8952	047160		047334	13\$:			
	047160	013746		MOV		RECDAT,-(SP)	::SAVE RECDAT FOR TYPEOUT
	047164	104402		TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)
8953	047166	000207				RTS PC	
8954	047170		047336	14\$:			
	047170	013746		MOV		EXDAT3,-(SP)	::SAVE EXDAT3 FOR TYPEOUT
	047174	104403		TYPOS			::GO TYPE--OCTAL ASCII

	047176	003		.BYTE	3	::TYPE 3 DIGIT(S)
	047177	001		.BYTE	1	::TYPE LEADING ZEROS
8955	047200	000207			RTS PC	
8956	047202		047340	15\$:		
	047202	013746		MOV	CHR50,-(SP)	::SAVE CHR50 FOR TYPEOUT
	047206	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047210	002		.BYTE	2	::TYPE 2 DIGIT(S)
	047211	001		.BYTE	1	::TYPE LEADING ZEROS
8957	047212	000207			RTS PC	
8958	047214		047342	16\$:		
	047214	013746		MOV	CMR119,-(SP)	::SAVE CMR119 FOR TYPEOUT
	047220	104403		TYPOS		::GO TYPE--OCTAL ASCII
	047222	001		.BYTE	1	::TYPE 1 DIGIT(S)
	047223	001		.BYTE	1	::TYPE LEADING ZEROS
8959	047224	000207			RTS PC	
8960	047226		047344	17\$:		
	047226	013746		MOV	FAILAD,-(SP)	::SAVE FAILAD FOR TYPEOUT
	047232	104402		TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
8961	047234	000207			RTS PC	

8962			
8963			
8964	047236	047300	
8965	047240	047304	
8966	047242	047310	
8967	047244	047312	
8968	047246	047314	
8969	047250	047316	
8970	047252	047320	
8971	047254	047322	
8972	047256	047324	
8973	047260	047326	
8974	047262	047330	
8975	047264	047332	
8976	047266	047334	
8977	047270	047336	
8978	047272	047340	
8979	047274	047342	
8980	047276	047344	

PRTABL: CA210
AMR210
CHR157
CA2113
CHR80
CDR150
EXDAT6
CA121
EXDAT1
CM1513
CNT121
FLTPAT
RECDAT
EXDAT3
CHR50
CMR119
FAILAD

8981			
8982	047300	000000	CA210: .WORD 0
8983	047302	000000	.WORD 0
8984	047304	000000	AMR210: .WORD 0
8985	047306	000000	.WORD 0
8986	047310	000000	CHR157: .WORD 0
8987	047312	000000	CA2113: .WORD 0
8988	047314	000000	CHR80: .WORD 0
8989	047316	000000	CDR150: .WORD 0
8990	047320	000000	EXDAT6: .WORD 0
8991	047322	000000	CA121: .WORD 0
8992	047324	000000	EXDAT1: .WORD 0
8993	047326	000000	CM1513: .WORD 0
8994	047330	000000	CNT121: .WORD 0
8995	047332	000000	FLTPAT: .WORD 0
8996	047334	000000	RECDAT: .WORD 0
8997	047336	000000	EXDAT3: .WORD 0
8998	047340	000000	CHR50: .WORD 0
8999	047342	000000	CMR119: .WORD 0

9000 047344 000000
9001
9002
9003
9004
9005
9006
9007 060000
9008 060000 000000
9009 070000
9010 070000 000000
9011 000001

FAILAD: .WORD 0

::*****

LOW1: .=60000
.WORD 0
HIGH1: .=70000
.WORD 0
.END

ABASE = 000000	BEBA - 170004	FAIL1 002060	SWR 002064	T145 025030
ACDW1 - 000000	BECC = 170002	FAIL2 002062	SWREG 000176	T146 025212
ACDW2 = 000000	BECR1 = 170006	FC = 000400	TDAR = 000001	T147 025532
ACPUOP= 000000	BECR2 = 170016	FIN 045570	TITL 045320	T15 003646
ADDW0 = 000000	BEDA = 170000	FLTPAT 047332	TPB = 001000	T150 026052
ADDW1 = 000000	BEGIN 002400	FMHI = 000010	TPE = 000040	T151 026446
ADDW10= 000000	BIT00 = 000001	FML0 = 000004	TSTCNT 002130	T152 027042
ADDW11= 000000	BIT01 = 000002	HIGH1 070000	TSTID = 000230	T153 027204
ADDW12= 000000	BIT02 = 000004	HIT = 000400	TSTIMS 002132	T154 027376
ADDW13= 000000	BIT03 = 000010	HODO = 000002	TYPDS = 104405	T155 027562
ADDW14= 000000	BIT04 = 000020	HPB = 004000	TYPE = 104401	T156 027742
ADDW15= 000000	BIT05 = 000040	HT = 000011	TYPOC = 104402	T157 030070
ADDW2 - 000000	BIT06 = 000100	INHERR 045556	TYPON = 104404	T16 003716
ADDW3 = 000000	BIT07 = 000200	KPAR0 = 172340	TYPOS = 104403	T160 030244
ADDW4 = 000000	BIT08 = 000400	KPAR1 = 172342	T1 002476	T161 030410
ADDW5 - 000000	BIT09 = 001000	KPAR2 = 172344	T10 003264	T162 030640
ADDW6 = 000000	BIT10 = 002000	KPAR3 = 172346	T100 012632	T163 031070
ADDW7 = 000000	BIT11 = 004000	KPAR4 = 172350	T101 013056	T164 031324
ADDW8 - 000000	BIT12 = 010000	KPAR5 = 172352	T102 013240	T165 031560
ADDW9 = 000000	BIT13 = 020000	KPAR6 = 172354	T103 013556	T166 032034
ADEVCT= 000000	BIT14 = 040000	KPAR7 = 172356	T104 014334	T167 032240
ADEVM = 000000	BIT15 = 100000	KPDR0 = 172300	T105 014546	T17 003766
ADRJMP 002124	CA121 047322	KPDR1 = 172302	T106 014700	T170 032342
ADRSYN 002122	CA210 047300	KPDR2 = 172304	T107 015036	T171 032542
ADR1\$ 002126	CA2113 047312	KPDR3 = 172306	T11 003326	T172 033100
AENV 000000	CCR = 177746	KPDR4 = 172310	T110 015222	T173 033436
AENVM - 000000	CDR = 177754	KPDR5 = 172312	T111 015420	T174 033726
AFATAL= 000000	CDR150 047316	KPDR6 = 172314	T112 015620	T175 034216
AM = 000010	CHR = 177752	KPDR7 = 172316	T113 016016	T176 034574
AMADR1- 000000	CHRPAT 002056	LF = 000012	T114 016216	T177 035226
AMADR2= 000000	CHR157 047310	LOOP 002052	T115 016440	T2 002572
AMADR3= 000000	CHR50 047340	LOW1 060000	T116 016634	T20 004036
AMADR4= 000000	CHR80 047314	LPB = 002000	T117 017230	T200 035660
AMAMS1= 000000	CME = 177744	MAGPRE 006774	T12 003406	T201 036214
AMAMS2= 000000	CMPE = 100000	OFF = 001015	T120 017324	T202 036550
AMAMS3= 000000	CMR - 177750	PEA = 000200	T121 017402	T203 037104
AMAMS4- 000000	CMRPAT 002054	PEHI = 000200	T122 017510	T204 037440
AMR210 047304	CMR119 047342	PELO = 000100	T123 017616	T205 037562
AMSGAD- 000000	CM1 - 100000	PRDATA 046714	T124 020034	T206 037704
AMSGLG- 000000	CM1513 047326	PRTABL 047236	T125 020236	T207 040026
AMSGTY- 000000	CM2 - 040000	PRTITL 045742	T126 020454	T21 004116
AMTYP1= 000000	CM3 = 020000	PSW = 177776	T127 020656	T210 040212
AMTYP2- 000000	CNT121 047330	RECDAT 047334	T13 003502	T211 040376
AMTYP3- 000000	CR - 000015	RELCTH 002352	T130 021124	T212 040520
AMTYP4= 000000	CRLF = 000200	RELCTL 002324	T131 021402	T213 040650
APASS = 000000	DAT 045416	SAVR0 045304	T132 021660	T214 041016
APRIOR- 000000	DCPI = 000001	SAVR1 045306	T133 022076	T215 041230
APTCSU= 000040	DISPRE 000174	SAVR2 045310	T134 022300	T216 041346
APTENV= 000001	EHA = 000004	SAVR3 045312	T135 022516	T217 041464
APTSIZ- 000200	ENDPAS 044034	SAVR4 045314	T136 022720	T22 004162
APTSPO= 000100	ERROR - 104000	SAVR5 045316	T137 023166	T220 041746
ASWREG= 000000	ERRPC 045250	SCPCND= 000004	T14 003552	T221 042114
ATESTN= 000000	ESA = 000020	SRO = 177572	T140 023444	T222 042644
AUNIT - 000000	EXDAT1 047324	SR3 = 172516	T141 023722	T223 043240
AUSWR - 000000	EXDAT3 047336	START 001000	T142 024146	T224 043404
AVECT1= 000000	EXDAT6 047320	STRTLP 002120	T143 024356	T225 043552
AVECT2- 000000	FAILAD 047344	STRTST 002116	T144 024556	T226 043706

T227	044010	T60	006554	\$ATYC	001630	\$ENV	001500	\$OMODE	044752
T23	004232	T61	006624	\$ATY1	001604	\$ENVM	001501	\$PASS	001466
T24	004302	T62	006674	\$ATY3	001612	\$ERROR	045252	\$PASTM	001452
T25	004352	T63	007172	\$ATY4	001622	\$ETABL	001500	\$QUES	002106
T26	004422	T64	007260	\$BASE	001534	\$ETEND	001604	\$SCPSE	002134
T27	004472	T65	007456	\$BELL	002102	\$FATAL	001462	\$SWR =	160000
T3	002666	T66	007636	\$CDW1	001540	\$FFLG	002050	\$SWREG	001502
T30	004542	T67	010102	\$CDW2	001542	\$FILLC	002100	\$TFSTN	001464
T31	004612	T7	003222	\$CHARC	044522	\$FILLS	002077	\$TIMES	044170
T32	004662	T70	010460	\$CPUOP	001506	\$HD =	000003	\$TKB	002070
T33	004732	T71	010626	\$CRLF	002107	\$HIBTS	001444	\$TKS	002066
T34	005002	T72	011156	\$DBLK	045170	\$ICNT	044172	\$TN =	000001
T35	005052	T73	011344	\$DDW0	001544	\$LF	002110	\$TPB	002074
T36	005122	T74	011536	\$DDW1	001546	\$LFLG	002047	\$TPFLG	002101
T37	005172	T75	011746	\$DDW10	001570	\$MADR1	001512	\$TPS	002072
T4	002762	T76	012166	\$DDW11	001572	\$MADR2	001516	\$TRAP	045200
T40	005242	T77	012412	\$DDW12	001574	\$MADR3	001522	\$TRAP2	045222
T41	005312	JCB =	001000	\$DDW13	001576	\$MADR4	001526	\$TRP =	000006
T42	005362	UMPRO0=	170200	\$DDW14	001600	\$MAIL	001460	\$TRPAD	045234
T43	005432	UMPRO1=	170202	\$DDW15	001602	\$MAMS1	001510	\$STSM	001450
T44	005502	UMPRO2=	170204	\$DDW2	001550	\$MAMS2	001514	\$TYPDS	044754
T45	005552	UMPRO3=	170206	\$DDW3	001552	\$MAMS3	001520	\$TYPE	044174
T46	005622	UMPRO4=	170210	\$DDW4	001554	\$MAMS4	001524	\$TYPEC	044406
T47	005672	UMPRO5=	170212	\$DDW5	001556	\$MBADR	001446	\$TYPEX	044524
T5	003056	UMPRO6=	170214	\$DDW6	001560	\$MFLG	002046	\$TYPOC	044552
T50	005750	UMPRO7=	170216	\$DDW7	001562	\$MSGAD	001474	\$TYPON	044566
T51	006026	UMPRO8=	170220	\$DDW8	001564	\$MSGLG	001476	\$TYPOS	044526
T52	006104	UMPRO9=	170222	\$DDW9	001566	\$MSGTY	001460	\$UNIT	001472
T53	006162	VCIP =	010000	\$DEVCT	001470	\$MTYP1	001511	\$UNITM	001454
T54	006240	VLD =	010000	\$DEVM	001536	\$MTYP2	001515	\$USWR	001504
T55	006316	VSIU =	020000	\$DOAGN	044164	\$MTYP3	001521	\$VECT1	001530
T56	006406	WWPD =	000100	\$DTBL	045160	\$MTYP4	001525	\$VECT2	001532
T57	006504	WWPT	002000	\$ENDAD	044154	\$NULL	002076	\$OFILL	044751
T6	003152	\$APTHD	001444	\$ENULL	002112	\$OCNT	044750	.\$X	001444

. ABS. 070002 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 63523 WORDS (249 PAGES)

DYNAMIC MEMORY: 20434 WORDS (78 PAGES)

ELAPSED TIME: 00:12:30

CKKAAO,CKKAAO/NL:TOC/CRF/-SP=CKKAAO.SML,CKKAAO.P11

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ABASE	=	000000	70-619 70-619
ACDW1	=	000000	70-619 70-619
ACDW2	=	000000	70-619 70-619
ACPUOP	=	000000	70-619 70-619
ADDW0	=	000000	70-619 70-619
ADDW1	=	000000	70-619 70-619
ADDW10	=	000000	70-619 70-619
ADDW11	=	000000	70-619 70-619
ADDW12	=	000000	70-619 70-619
ADDW13	=	000000	70-619 70-619
ADDW14	=	000000	70-619 70-619
ADDW15	=	000000	70-619 70-619
ADDW2	=	000000	70-619 70-619
ADDW3	=	000000	70-619 70-619
ADDW4	=	000000	70-619 70-619
ADDW5	=	000000	70-619 70-619
ADDW6	=	000000	70-619 70-619
ADDW7	=	000000	70-619 70-619
ADDW8	=	000000	70-619 70-619
ADDW9	=	000000	70-619 70-619
ADEVCT	=	000000	70-619 70-619
ADEVN	=	000000	70-619 70-619
ADRJMP		002124	#72-775 *72-810 72-812 72-814 90-8835 90-8839
ADRSYN		002122	#72-774 *72-809 90-8841
ADR1\$		002126	#72-776 *72-812 *72-813 72-815 90-8836 90-8840
AENV		000000	70-619 70-619
AENVN	=	000000	70-619 70-619
AFATAL	=	000000	70-619 70-619
AM		000010	#71-739 75-1755 75-1860 77-1919 77-1959 77-2003 77-2060 77-2133 77-2188
			77-2246
AMADR1		000000	70-619 70-619
AMADR2		000000	70-619 70-619
AMADR3	=	000000	70-619 70-619
AMADR4	=	000000	70-619 70-619
AMAMS1	=	000000	70-619 70-619
AMAMS2	=	000000	70-619 70-619
AMAMS3	=	000000	70-619 70-619
AMAMS4	=	000000	70-619 70-619
AMR210		047304	*77-1963 *77-1965 *77-1967 *77-1968 77-1976 *77-2007 *77-2008 *77-2009 77-2015
			*77-2062 *77-2064 77-2071 *77-2138 *77-2140 *77-2144 77-2151 *77-2190 *77-2192
			77-2201 *77-2250 *77-2253 *77-2256 *77-2258 77-2267 90-8929 90-8930 90-8965
			#90-8984
AMSGAD	=	000000	70-619 70-619
AMSGLG	=	000000	70-619 70-619
AMSGTY	=	000000	70-619 70-619
AMTYP1	=	000000	70-619 70-619
AMTYP2	=	000000	70-619 70-619
AMTYP3	=	000000	70-619 70-619
AMTYP4	=	000000	70-619 70-619
APASS	=	000000	70-619 70-619
APRIOR	=	000000	70-619
APTCSU	=	000040	#70-620 #71-721 90-8740

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
APTENV	=	000001	70-620 #70-620 #71-719 90-8740 90-8823
APTSIZ	=	000200	69-605 #70-620 #71-718 79-2764 88-8345 89-8447
APTSP0	=	000100	70-620 #70-620 #71-720 90-8740
ASWREG	=	000000	70-619 70-619
ATESTN	=	000000	70-619 70-619
AUNIT	=	000000	70-619 70-619
AUSWR	=	000000	70-619 70-619
AVECT1	=	000000	70-619 70-619
AVECT2	=	000000	70-619 70-619
BEBA	=	170004	#71-713 *89-8384 *89-8471
BECC	=	170002	#71-712 *89-8386 *89-8473
BECR1	=	170006	#71-715 88-8353 *89-8389 89-8490 89-8455 *89-8476 89-8477
BECR2	=	170016	#71-716 *89-8388 *89-8475 89-848
BEDA	=	170000	#71-714 *89-8387 *89-8474
BEGIN	=	002400	69-616 #73-847 89-8730
BIT00	=	000001	#71-666 74-1077 74-1080 75-1451 75-1453 75-1650 75-1654 75-1670 75-1674
BIT01	=	000002	#71-667 75-1303 75-1305 75-1462 75-1464 75-1690 75-1694 75-1710 75-1714
BIT02	=	000004	#71-668 74-1097 74-1101 74-1120 75-1473 75-1475 75-1598 75-1599 75-1601
			75-1619 75-1620
BIT03	=	000010	#71-669 74-1139 74-1143 74-1163 75-1484 75-1486 75-1736 75-1759
BIT04	=	000020	#71-670 75-1325 75-1327 75-1495 75-1497
BIT05	=	000040	#71-671 75-1345 75-1347 75-1786 75-1788
BIT06	=	000100	#71-672 74-1180 74-1182 75-1797 75-1799
BIT07	=	000200	#71-673 74-1201 74-1203 74-1220 74-1224 75-1808 75-1810
BIT08	=	000400	#71-674 74-1076 75-1242 75-1244 75-1365 75-1367 75-1506 75-1508 82-4651
BIT09	=	001000	#71-675 75-1262 75-1517 75-1519 90-8830
BIT10	=	002000	#71-676 75-1277 75-1279 75-1528 75-1530 75-1622
BIT11	=	004000	#71-677 72-798 75-1385 75-1387 75-1539 75-1541 89-8711
BIT12	=	010000	#71-678 74-1056 75-1550 75-1552
BIT13	=	020000	#71-679 75-1561 75-1563 90-8765
BIT14	=	040000	#71-680 72-786 75-1403 75-1405 75-1572 75-1574
BIT15	=	100000	#71-681 75-1423 75-1425 90-8827
CA121	=	047322	*79-2910 *79-2911 79-2916 *80-3273 *80-3274 80-3279 *80-3615 *80-3617 80-3622
			*80-3665 *80-3667 80-3672 *80-3722 *80-3724 80-3729 *80-3772 *80-3774 80-3779
			*80-3832 *80-3833 80-3841 *81-4046 *81-4048 81-4053 *81-4096 *81-4098 81-4103
			*81-4153 *81-4155 81-4160 *81-4203 *81-4205 81-4210 *81-4263 *81-4264 81-4272
			*82-4798 *82-4799 82-4814 *82-4824 *82-4825 82-4837 *82-4905 *82-4906 82-4921
			*82-4931 *82-4932 83-4945 *83-5027 *83-5028 83-5043 *83-5061 *83-5062 83-5075
			*83-5152 *83-5153 83-5168 *83-5186 *83-5187 83-5200 *84-5677 *84-5678 84-5686
			84-5698 84-5707 *84-5760 *84-5761 84-5769 84-5780 84-5789 *84-5845 *84-5846
			84-5854 84-5866 84-5875 *84-5928 *84-5929 84-5937 84-5948 84-5957 *84-6031
			*84-6032 84-6040 *84-6051 *84-6052 84-6060 *84-6120 *84-6121 84-6129 *86-6826
			*86-6827 86-6842 *86-6859 *86-6860 86-6872 *86-6964 *86-6965 86-6980 *86-6997
			*86-6998 86-7010 *87-7101 *87-7102 87-7114 *87-7123 *87-7124 87-7133 *87-7213
			*87-7214 87-7226 *87-7235 *87-7236 87-7245 *87-7334 *87-7335 87-7347 *87-7356
			*87-7357 87-7366 *87-7446 *87-7447 87-7459 *87-7468 *87-7469 87-7478 *89-8409
			*89-8411 89-8418 90-8942 90-8971 #90-8991
CA210	=	047300	*77-1961 *77-1962 77-1975 *77-2005 *77-2006 77-2014 *77-2065 *77-2066 77-2070
			*77-2139 *77-2141 *77-2143 77-2150 *77-2193 *77-2194 77-2199 *77-2252 *77-2254
			*77-2257 *77-2259 77-2265 *77-2446 *77-2447 77-2458 *77-2497 *77-2498 77-2508
			*77-2547 *77-2548 78-2560 *79-2600 *79-2601 79-2611 *79-3079 *79-3080 79-3085
			*79-3122 *79-3123 79-3128 *79-3167 *79-3168 79-3174 *79-3212 *79-3213 80-3219

SYMBOL CROSS REFERENCE
SYMBOL VALUE

REFERENCES

		*87-7591	*87-7601	*87-7639	*87-7645	*87-7654	*87-7691	*87-7696	*87-7699	*88-7705
		*88-7714	*88-7716	88-7718	*88-7756	*88-7762	*88-7765	*88-7771	*88-7782	*88-7784
		88-7786	*88-7827	*88-7833	*88-7841	*88-7883	*88-7889	*88-7893	*88-7897	*88-7941
		*88-7949	*88-7954	*88-7958	*88-7966	*88-8007	*88-8022	*88-8040	*88-8082	*88-8087
		*88-8099	*88-8141	*88-8146	*88-8159	*88-8191	*88-8207	*88-8240	*88-8275	*88-8281
		*88-8294	*88-8336	88-8337	*88-8339	88-8340	*89-8372	*89-8404	*89-8438	89-8439
		*89-8441	89-8442	*89-8467	89-8483	*89-8492	*89-8528	*89-8534	*89-8546	*89-8572
		*89-8578	*89-8590	*89-8641						
CDR	- 177754	#71-656	74-983	79-2960	79-2995	79-3031	79-3073	79-3116	79-3161	79-3206
		80-3266	81-4459	81-4465	82-4784	82-4789	82-4891	82-4896		
CDR150	047316	*79-2960	79-2964	79-2970	*79-2995	79-2999	79-3005	*79-3031	79-3035	79-3043
		*79-3073	79-3077	79-3086	*79-3116	79-3120	79-3129	*79-3161	79-3165	79-3175
		*79-3206	79-3210	80-3220	*81-4473	81-4480	*81-4485	81-4492	*82-4796	82-4812
		*82-4822	82-4835	*82-4903	82-4919	*82-4929	83-4943	90-8938	90-8969	#90-8989
CHR	177752	#71-655	74-957	*75-1730	*75-1752	*75-1841	*75-1844	*77-1913	*77-1944	*77-1996
		*77-2044	*77-2108	*77-2176	*77-2232	*77-2298	77-2310	*77-2338	77-2350	*77-2380
		77-2392	*77-2425	77-2439	*77-2476	77-2490	*77-2526	77-2540	*79-2579	79-2593
		79-2653	79-2712	79-2803	79-2903	*80-3590	*80-3697	*81-4021	*81-4128	*81-4510
		81-4526	82-4577	*83-4983	*83-5001	83-5008	83-5011	*83-5108	*83-5126	83-5133
		83-5136	*83-5498	*83-5551	*84-5602	*84-5654	*84-5737	*84-5820	*84-5905	*86-6764
		*86-6785	*86-6902	*86-6923	89-8544	89-8588	*90-8841			
CHRPAT	002056	#70-628	*77-1942	77-1944	77-1963	77-1965	*77-1979	*77-2103	77-2108	77-2138
		77-2139	*77-2159	*77-2220	77-2232	*77-2272				
CHR157	047310	*77-2310	*77-2314	77-2315	*77-2318	*77-2319	77-2326	*77-2350	*77-2354	77-2355
		*77-2358	*77-2359	77-2366	*77-2392	*77-2397	*77-2398	77-2401	77-2408	*77-2439
		*77-2443	77-2444	*77-2449	*77-2450	77-2457	*77-2490	*77-2494	77-2495	*77-2500
		*77-2501	77-2509	*77-2540	*77-2544	77-2545	*77-2550	*77-2551	78-2559	*79-2593
		*79-2597	79-2598	*79-2603	*79-2604	79-2612	*79-2653	*79-2657	79-2658	*79-2661
		*79-2662	79-2669	*79-2712	*79-2718	79-2719	*79-2725	*79-2727	79-2736	*79-2803
		*79-2810	79-2811	*79-2817	*79-2819	79-2828	*81-4526	*82-4531	82-4532	*82-4535
		*82-4536	82-4548	*82-4577	*82-4581	82-4582	*82-4585	*82-4586	82-4595	*83-5019
		*83-5022	*83-5023	83-5041	*83-5053	*83-5056	*83-5057	83-5074	*83-5144	*83-5147
		*83-5148	83-5166	*83-5178	*83-5181	*83-5182	83-5199	90-8932	90-8966	#90-8986
CHR50	047340	*89-8553	89-8557	*89-8597	89-8601	90-8956	90-8978	#90-8998		
CHR80	047314	*77-2378	77-2380	77-2401	77-2409	*77-2412	77-2413	90-8936	90-8968	#90-8988
CME	177744	#71-652	74-879	*74-1009	74-1011	*75-1451	75-1453	*75-1462	75-1464	*75-1473
		75-1475	*75-1484	75-1486	*75-1495	75-1497	*75-1506	75-1508	*75-1517	75-1519
		*75-1528	75-1530	*75-1539	75-1541	*75-1550	75-1552	*75-1561	75-1563	*75-1572
		75-1574	*84-6169	84-6170	*84-6215	84-6224	*85-6296	85-6302	85-6307	*85-6310
		85-6311	*85-6413	85-6420	85-6425	*85-6428	85-6429	*86-6524	*86-6592	*86-6670
		86-6690	*86-6691	86-6692	*88-7706	*88-7772				
CMPE	- 100000	#71-754	86-6703	86-6711						
CMR	177750	#71-654	74-931	*75-1650	75-1651	*75-1653	*75-1670	75-1671	*75-1673	*75-1690
		75-1691	*75-1693	*75-1710	75-1711	*75-1713	*75-1731	*75-1732	75-1733	*75-1735
		*75-1753	*75-1754	*75-1755	75-1756	*75-1758	*75-1786	75-1788	*75-1797	75-1799
		*75-1808	75-1810	*75-1842	*75-1843	*75-1845	75-1858	*77-1914	*77-1915	77-1919
		*77-1945	*77-1946	77-1957	*77-1995	*77-1997	77-2001	*77-2045	*77-2046	77-2056
		*77-2107	*77-2111	77-2129	*77-2177	*77-2178	77-2188	*77-2231	*77-2235	77-2246
		*77-2299	*77-2312	*77-2339	*77-2352	*77-2382	*77-2394	*77-2429	*77-2441	*77-2480
		*77-2492	*77-2530	*77-2542	*79-2583	*79-2595	*79-2643	*79-2655	*79-2697	*79-2717
		*79-2792	*79-2809	*79-2891	*79-2905	*79-2950	*79-2962	*79-2985	*79-2997	*79-3021
		*79-3033	*79-3066	*79-3075	*79-3109	*79-3118	*79-3154	*79-3163	*79-3199	*79-3208

		REFERENCES								
		*80-3253	*80-3268	*80-3319	80-3330	*80-3332	*80-3392	80-3409	*80-3414	*80-3446
		*80-3592	*80-3596	80-3608	*80-3610	*80-3649	80-3658	*80-3660	*80-3699	*80-3703
		80-3715	*80-3717	*80-3756	80-3765	*80-3767	*80-3807	*80-3811	*80-3816	80-3824
		*80-3826	*81-3880	81-3903	*81-3906	*81-3948	81-3971	*81-3974	*81-4023	*81-4027
		81-4039	*81-4041	*81-4080	81-4089	*81-4091	*81-4130	*81-4134	81-4146	*81-4148
		*81-4187	81-4196	*81-4198	*81-4238	*81-4242	*81-4247	81-4255	*81-4257	*81-4310
		81-4333	*81-4336	*81-4375	81-4398	*81-4401	*81-4447	*81-4467	*81-4511	*81-4518
		*82-4530	*82-4561	*82-4579	*82-4628	82-4642	82-4646	*82-4649	*82-4707	82-4721
		*82-4724	*82-4768	*82-4794	*82-4820	*82-4844	*82-4875	*82-4901	*82-4927	*83-4952
		*83-4987	*83-5017	*83-5051	*83-5082	*83-5112	*83-5142	*83-5176	*83-5207	*83-5246
		83-5257	*83-5259	*83-5303	83-5315	*83-5317	*83-5369	83-5381	*83-5383	*83-5430
		83-5442	*83-5445	*83-5499	83-5512	*83-5514	*83-5553	83-5566	*83-5568	*84-5603
		84-5617	*84-5620	*84-5661	84-5673	*84-5675	*84-5744	84-5756	*84-5758	*84-5829
		84-5841	*84-5843	*84-5912	84-5924	*84-5926	*84-6008	84-6023	*84-6025	*84-6098
		84-6113	*84-6115	*84-6164	*84-6172	*84-6207	*84-6227	*85-6284	*85-6314	*85-6400
		*85-6432	*86-6511	*86-6539	*86-6579	*86-6606	*86-6656	*86-6695	*86-6771	*86-6787
		86-6795	*86-6796	86-6808	*86-6816	*86-6850	*86-6879	*86-6909	*86-6925	86-6933
		*86-6934	86-6946	*86-6954	*86-6988	*86-7017	*87-7061	87-7089	87-7092	*87-7100
		*87-7122	*87-7140	*87-7173	87-7201	87-7204	*87-7212	*87-7234	*87-7252	*87-7294
		87-7322	87-7325	*87-7333	*87-7355	*87-7373	*87-7406	87-7434	87-7437	*87-7445
		*87-7467	*87-7485	*87-7534	87-7545	*87-7587	87-7599	*87-7641	87-7652	*87-7693
		88-7713	*88-7759	88-7781	*88-7829	88-7839	*88-7885	88-7896	*88-7945	88-7964
		*88-8018	88-8033	88-8035	*88-8083	88-8097	*88-8142	88-8157	*88-8202	*88-8223
		*88-8239	*89-8373	*89-8381	*89-8392	89-8402	*89-8405	*89-8531	*89-8547	*89-8575
		*89-8591	*90-8842							
CMRPAT	002054	#70-627	*77-1993	77-1995	77-2008	*77-2017	77-2018	*77-2102	77-2107	*77-2158
		*77-2219	77-2231	*77-2271						
CMR119	047342	*86-6818	*86-6821	*86-6822	86-6840	*86-6852	*86-6855	*86-6856	86-6870	*86-6956
		*86-6959	*86-6960	86-6978	*86-6990	*86-6993	*86-6994	86-7008	90-8958	90-8979
		#90-8999								
CM1	- 100000	#71-748								
CM1513	047326	*80-3330	*80-3334	80-3335	*80-3341	*80-3342	80-3350	*80-3409	*80-3416	*80-3417
		80-3420	80-3436	90-8946	90-8973	#90-8993				
CM2	= 040000	#71-747								
CM3	- 020000	#71-746								
CNT121	047330	*81-3909	*81-3911	81-3918	*81-3977	*81-3979	81-3986	*81-4338	*81-4340	81-4347
		*81-4404	*81-4406	81-4413	90-8948	90-8974	#90-8994			
CR	- 000015	90-8740	90-8740							
CRLF	000200	69-595	69-596	69-597	69-598	69-614	79-2846	79-2847	89-8367	89-8367
		89-8369	89-8369	89-8720	90-8740	90-8740	90-8771	90-8771	90-8786	
DAT	045416	90-8774	#90-8786							
DCPI	= 000001	#71-724	86-6525							
DISPRE	000174	#68-557								
EHA	= 000004	#71-738								
ENDPAS	044034	#89-8709								
ERROR	= 104000	74-887	74-913	74-939	74-965	74-991	74-1013	74-1034	74-1059	74-1082
		74-1103	74-1122	74-1145	74-1165	74-1184	74-1205	74-1226	75-1264	75-1281
		75-1307	75-1329	75-1349	75-1369	75-1389	75-1407	75-1427	75-1455	75-1466
		75-1477	75-1488	75-1499	75-1510	75-1521	75-1532	75-1543	75-1554	75-1565
		75-1576	75-1603	75-1624	75-1656	75-1676	75-1696	75-1716	75-1738	75-1761
		75-1790	75-1801	75-1812	75-1862	77-1921	77-1971	77-2010	77-2067	77-2147
		77-2195	77-2263	77-2322	77-2362	77-2403	77-2453	77-2504	77-2554	79-2607

REFERENCES

		79-2665	79-2731	79-2823	79-2912	79-2966	79-3001	79-3038	79-3081	79-3124
		79-3169	79-3214	80-3275	80-3345	80-3430	80-3482	80-3505	80-3534	80-3562
		80-3618	80-3668	80-3725	80-3775	80-3834	81-3912	81-3980	81-4049	81-4099
		81-4156	81-4206	81-4265	81-4341	81-4407	81-4474	81-4486	82-4540	82-4590
		82-4656	82-4664	82-4673	82-4681	82-4727	82-4800	82-4826	82-4907	82-4933
		83-5029	83-5063	83-5154	83-5188	83-5263	83-5272	83-5321	83-5332	83-5388
		83-5400	83-5453	83-5463	83-5519	83-5572	84-5626	84-5681	84-5693	84-5703
		84-5764	84-5775	84-5785	84-5849	84-5861	84-5871	84-5932	84-5943	84-5953
		84-6033	84-6053	84-6122	84-6176	84-6235	85-6325	85-6337	85-6349	85-6443
		85-6455	85-6467	86-6549	86-6616	86-6705	86-6713	86-6721	86-6728	86-6828
		86-6861	86-6966	86-6999	87-7103	87-7125	87-7215	87-7237	87-7336	87-7358
		87-7448	87-7470	87-7551	87-7605	87-7658	88-7723	88-7791	88-7845	88-7902
		88-7970	88-8046	88-8103	88-8163	88-8246	88-8257	88-8300	89-8412	89-8496
		89-8505	89-8554	89-8598	89-8645	89-8690	#90-8747			
ERRPC	045250	#90-8748	*90-8769	90-8789						
ESA	- 000020	#71-740								
EXDAT1	047324	*80-3339	80-3349	*80-3429	80-3435	*86-6817	86-6839	*86-6851	86-6869	*86-6955
		86-6977	*86-6989	86-7007	90-8944	90-8972	#90-8992			
EXDAT3	047336	*82-4539	82-4547	*82-4589	82-4594	*83-5018	83-5040	*83-5052	83-5071	*83-5143
		83-5165	*83-5177	83-5196	90-8954	90-8977	#90-8997			
EXDAT6	047320	*79-3037	79-3042	*81-4472	81-4479	*81-4484	81-4491	*82-4795	82-4811	*82-4821
		82-4834	*82-4902	82-4918	*82-4928	83-4942	*84-6232	84-6238	*85-6323	85-6329
		*85-6335	85-6341	*85-6347	85-6353	*85-6441	85-6447	*85-6453	85-6459	*85-6465
		85-6471	*88-8299	88-8306	*89-8643	89-8647	90-8940	90-8970	#90-8990	
FAILAD	047344	*89-8642	89-8646	90-8960	90-8980	#90-9000				
FAIL1	002060	#70-629	*80-3472	*80-3476	80-3480	*86-6515	*86-6533	86-6547	*86-6583	*86-6603
		86-6614	*86-6660	*86-6685	86-6726	*88-8196	*88-8235	88-8244		
FAIL2	002062	#70-630	*88-8197	*88-8237	88-8254					
FC	= 000400	#71-729	80-3473	80-3499	80-3524	80-3527	80-3552	80-3555	80-3587	80-3643
		80-3694	80-3750	80-3799	81-3876	81-3890	81-3894	81-3944	81-3958	81-3962
		81-4018	81-4074	81-4125	81-4181	81-4230	81-4306	81-4320	81-4324	81-4371
		81-4385	81-4389	82-4624	82-4700	83-5446	84-5621	84-6203	85-6280	85-6316
		85-6395	85-6434	86-6504	86-6543	86-6572	86-6610	86-6648	86-6699	87-7053
		87-7165	87-7286	87-7398	88-7716	88-7784	88-8339	89-8441		
FIN	045570	90-8795	90-8813	#90-8818						
FLTPAT	047332	*77-2033	77-2044	77-2047	77-2062	77-2065	*77-2073	77-2074	*77-2094	77-2106
		77-2140	77-2141	*77-2154	77-2155	*79-2776	79-2788	*79-2832	*79-2884	79-2887
		79-2888	79-2910	*79-2922	79-2923	*79-3019	79-3020	79-3035	79-3037	*79-3046
		*80-3246	80-3249	80-3250	80-3273	*80-3285	80-3286	*80-3802	80-3805	80-3806
		80-3832	*80-3847	80-3848	*81-4233	81-4236	81-4237	81-4263	*81-4278	81-4279
		*83-5301	83-5302	83-5326	83-5337	*83-5341	83-5342	*83-5367	83-5368	83-5393
		83-5405	*83-5409	*83-5550	83-5551	83-5577	*83-5581	83-5582	*84-6001	84-6004
		84-6005	84-6031	84-6051	*84-6067	84-6068	*84-6093	84-6096	84-6097	84-6120
		*84-6136	84-6137	*88-8278	88-8279	88-8295	88-8299	*88-8311	90-8950	90-8975
		#90-8995								
FMHI	000010	#71-726	74-1161							
FMLO	000004	#71-725	74-1118	86-6525	86-6593	86-6671	88-7705	88-7771		
GNS	*****	68-557	68-557	69-595	69-596	69-597	69-598	69-614	79-2846	79-2847
		89-8367	89-8368	89-8369	89-8720	90-8743	90-8743	90-8743	90-8743	90-8743
		90-8743	90-8743	90-8743	90-8743	90-8743	90-8771	90-8786	90-8788	90-8801
		90-8804	90-8872	90-8874	90-8876	90-8878	90-8880	90-8882	90-8884	90-8886
		90-8888	90-8890	90-8892	90-8894	90-8896	90-8898	90-8900	90-8902	90-8904

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
HIGH1		070000	72-834 #90-9010
HIT	=	000400	#71-741 87-7548 87-7602 87-7655 88-7720 88-7788 88-7842 88-7899 88-7967
HODO	-	000002	88-8043 88-8100 88-8160
			#71-737 77-2299 77-2339 77-2382 77-2429 77-2480 77-2530 79-2583 79-2643
			79-2697 79-2792 79-2891 79-2950 79-2985 79-3021 79-3066 79-3109 79-3154
			79-3199 80-3253 80-3319 80-3392 80-3596 80-3649 80-3703 80-3756 80-3807
			81-3880 81-3948 81-4027 81-4080 81-4134 81-4187 81-4238 81-4310 81-4375
			81-4447 81-4511 82-4561 82-4628 82-4707 82-4768 82-4875 83-4987 83-5112
			83-5246 83-5303 83-5369 83-5430 83-5499 83-5553 84-5603 84-5661 84-5744
			84-5829 84-5912 84-6008 84-6098 84-6164 84-6207 85-6284 85-6400 86-6511
			86-6579 86-6656 86-6771 86-6909 87-7061 87-7173 87-7294 87-7406 87-7534
			87-7587 87-7641 87-7693 88-7759 88-7829 88-7885 88-7945 88-8018 88-8083
			88-8142 88-8202 88-8223 89-8373 89-8381 89-8392 89-8531 89-8575
HPB		004000	#71-744 83-5261 83-5319 83-5385 83-5450 84-5679 84-5762 84-5847 84-5930
			84-6027 84-6047
HT	=	000011	90-8740 90-8740
INHERR	=	045556	90-8766 #90-8810
KPAR0	=	172340	#71-692 *75-1881
KPAR1	=	172342	#71-693 *75-1882
KPAR2	=	172344	#71-694 *75-1883
KPAR3	=	172346	#71-695 *75-1884
KPAR4	=	172350	#71-696 *77-2036 *77-2090 *77-2222 77-2253 77-2254 *77-2269 *79-2689 79-2719
			79-2723 *79-2741 *79-2774 79-2811 79-2815 *79-2831 *80-3382 80-3423 *80-3438
			*88-8010
KPAR5	-	172352	#71-697
KPAR6	=	172354	#71-698
KPAR7	=	172356	#71-699 *75-1885
KPDR0	=	172300	#71-684 *75-1873
KPDR1	=	172302	#71-685 *75-1874
KPDR2	=	172304	#71-686 *75-1875
KPDR3	=	172306	#71-687 *75-1876
KPDR4	=	172310	#71-688 *75-1877
KPDR5	-	172312	#71-689 *75-1878
KPDR6	-	172314	#71-690 *75-1879
KPDR7	=	172316	#71-691 *75-1880
LF	=	000012	90-8740 90-8740
LOOP		002052	#70-626 *73-858 *73-859 *77-1966 *77-1969 *77-2142 *77-2145 *77-2255 *77-2260
			*77-2317 *77-2320 *77-2357 *77-2360 *77-2396 *77-2399 *77-2448 *77-2451 *77-2499
			*77-2502 *77-2549 *77-2552 *79-2602 *79-2605 *79-2660 *79-2663 *79-2724 *79-2729
			*79-2816 *79-2821 *80-3340 *80-3343 *80-3415 *80-3418 *80-3424 *80-3427 *82-4534
			*82-4537 *82-4584 *82-4587 *83-5021 *83-5024 *83-5055 *83-5058 *83-5146 *83-5149
			*83-5180 *83-5183 *86-6820 *86-6823 *86-6854 *86-6857 *86-6958 *86-6961 *86-6992
			*86-6995 *89-8479 *89-8480
LOW1		060000	72-824 #90-9008
LPB		002000	#71-743 83-5270 83-5330 83-5397 83-5460 84-5690 84-5773 84-5858 84-5941
MAGPRE		006774	#75-1873
OFF	-	001015	69-589 #71-758 73-850 74-1076 74-1096 74-1100 74-1138 74-1142 74-1223
			77-1941 77-1992 77-2032 77-2089 77-2174 77-2218 77-2297 77-2313 77-2337
			77-2353 77-2377 77-2395 77-2424 77-2442 77-2475 77-2493 77-2525 77-2543
			79-2578 79-2596 79-2642 79-2656 79-2688 79-2716 79-2763 79-2808 79-2883
			79-2906 79-2948 79-2963 79-2983 79-2998 79-3018 79-3034 79-3059 79-3076
			79-3102 79-3119 79-3147 79-3164 79-3192 79-3209 80-3241 80-3269 80-3318

		REFERENCES								
		80-3333	80-3381	80-3413	80-3447	80-3584	80-3611	80-3640	80-3661	80-3691
		80-3718	80-3747	80-3768	80-3796	80-3827	80-3870	81-3907	81-3939	81-3975
		81-4015	81-4042	81-4071	81-4092	81-4122	81-4149	81-4178	81-4199	81-4227
		81-4258	81-4301	81-4337	81-4366	81-4402	81-4442	81-4468	81-4509	82-4528
		82-4560	82-4580	82-4621	82-4647	82-4697	82-4722	82-4759	82-4793	82-4819
		82-4843	82-4866	82-4900	82-4926	83-4951	83-4981	83-5016	83-5050	83-5081
		83-5106	83-5141	83-5175	83-5206	83-5244	83-5260	83-5300	83-5318	83-5366
		83-5384	83-5428	83-5444	83-5497	83-5515	83-5549	83-5569	84-5601	84-5619
		84-5653	84-5676	84-5736	84-5759	84-5819	84-5844	84-5904	84-5927	84-5995
		84-6026	84-6092	84-6116	84-6163	84-6202	84-6226	85-6279	85-6313	85-6394
		85-6431	86-6503	86-6534	86-6536	86-6571	86-6600	86-6602	86-6647	86-6686
		86-6688	86-6762	86-6815	86-6849	86-6878	86-6900	86-6953	86-6987	86-7016
		87-7050	87-7099	87-7121	87-7139	87-7162	87-7211	87-7233	87-7251	87-7283
		87-7332	87-7354	87-7372	87-7395	87-7444	87-7466	87-7484	87-7532	87-7547
		87-7585	87-7601	87-7639	87-7654	87-7691	88-7714	88-7756	88-7782	88-7827
		88-7841	88-7883	88-7897	88-7941	88-7966	88-8007	88-8040	88-8082	88-8099
		88-8141	88-8159	88-8191	88-8240	88-8275	88-8294	88-8336	89-8404	89-8438
		89-8492	89-8528	89-8546	89-8572	89-8590	89-8641			
PEA	= 000200	#71-728	84-6220	85-6303	85-6421	86-6672				
PEHI	= 000200	#71-753								
PELO	= 000100	#71-752								
PRDATA	046714	90-8797	#90-8909							
PRTABL	047236	90-8775	90-8796	#90-8964						
PRTITL	045742	90-8776	#90-8855							
PSW	- 177776	#71-657	*72-783	*73-849	*88-8201	*89-8484				
RECDAT	047334	*84-6175	84-6179	*84-6233	84-6239	*85-6324	85-6330	*85-6336	85-6342	*85-6348
		85-6354	*85-6442	85-6448	*85-6454	85-6460	*85-6466	85-6472	*88-8256	88-8262
RELCTH	002352	*88-8298	88-8308	*89-8644	89-8648	90-8952	90-8976	#90-8996		
		#72-834	74-1096	77-2297	77-2337	77-2377	77-2424	77-2475	79-2642	79-2688
		79-2763	79-2883	79-2948	79-2983	79-3018	79-3059	79-3102	80-3241	80-3318
		80-3381	80-3584	80-3640	80-3796	80-3870	81-4015	81-4071	81-4227	81-4301
		81-4442	81-4509	82-4621	82-4697	82-4759	83-4981	83-5244	83-5300	83-5366
		83-5428	83-5497	83-5549	84-5601	84-5653	84-5819	84-5995	84-6092	84-6163
		84-6202	85-6279	85-6394	86-6503	86-6571	86-6647	86-6762	87-7050	87-7283
		87-7532	87-7639	87-7691	88-7756	88-7883	88-7941	88-8007	88-8082	88-8141
RELCTL	002324	88-8191	88-8275	88-8336	89-8438	89-8528	89-8572			
		#72-824	74-1138	77-1941	77-1992	77-2032	77-2089	77-2174	77-2218	77-2525
		79-2578	79-3147	79-3192	80-3691	80-3747	81-3939	81-4122	81-4178	81-4366
		82-4560	82-4866	83-5106	84-5736	84-5904	86-6900	87-7162	87-7395	87-7585
		88-7827								
SAVRO	045304	*90-8750	#90-8757	90-8847						
SAVR1	045306	*90-8751	#90-8758	90-8848						
SAVR2	045310	*90-8752	#90-8759	90-8849						
SAVR3	045312	*90-8753	#90-8760	90-8850						
SAVR4	045314	*90-8754	#90-8761	90-8851						
SAVR5	045316	*90-8755	#90-8762	90-8852						
SCPCND	- 000004	#71-757	74-876	74-902	74-928	74-954	74-980	74-1008	74-1030	74-1055
		74-1075	74-1096	74-1117	74-1138	74-1160	74-1179	74-1200	74-1219	75-1241
		75-1259	75-1276	75-1302	75-1324	75-1344	75-1364	75-1384	75-1402	75-1422
		75-1450	75-1461	75-1472	75-1483	75-1494	75-1505	75-1516	75-1527	75-1538
		75-1549	75-1560	75-1571	75-1597	75-1618	75-1649	75-1669	75-1689	75-1709
		75-1729	75-1751	75-1785	75-1796	75-1807	75-1840	77-1912	77-1941	77-1992

CKKAAO CREATED BY MACRO ON 26-SEP-79 AT 09:01
 SYMBOL CROSS REFERENCE
 SYMBOL VALUE

REFERENCES

PAGE 9
 CREF

N 3

SEQ 0246

		77-2032	77-2089	77-2174	77-2218	77-2297	77-2337	77-2377	77-2424	77-2475
		77-2525	79-2578	79-2642	79-2688	79-2763	79-2883	79-2948	79-2983	79-3018
		79-3059	79-3102	79-3147	79-3192	80-3241	80-3318	80-3381	80-3470	80-3497
		80-3521	80-3549	80-3584	80-3640	80-3691	80-3747	80-3796	80-3870	81-3939
		81-4015	81-4071	81-4122	81-4178	81-4227	81-4301	81-4366	81-4442	81-4509
		82-4560	82-4621	82-4697	82-4759	82-4866	83-4981	83-5106	83-5244	83-5300
		83-5366	83-5428	83-5497	83-5549	84-5601	84-5653	84-5736	84-5819	84-5904
		84-5995	84-6092	84-6163	84-6202	85-6279	85-6394	86-6503	86-6571	86-6647
		86-6762	86-6900	87-7050	87-7162	87-7283	87-7395	87-7532	87-7585	87-7639
		87-7691	88-7756	88-7827	88-7883	88-7941	88-8007	88-8082	88-8141	88-8191
		88-8275	88-8336	89-8438	89-8528	89-8572	89-8626	89-8665	89-8700	
SRO	- 177572	#71-700	*77-2042	*77-2057	*77-2109	*77-2130	*77-2233	*77-2243	*79-2700	*79-2714
		*79-2779	*79-2806	*80-3396	*80-3411	*80-3444	*88-8015	*88-8038		
SR3	= 172516	#71-701	*77-2043	*77-2058	*77-2110	*77-2131	*77-2234	*77-2244	*79-2701	*79-2715
		*79-2780	*79-2807	*80-3397	*80-3412	*80-3445	*88-8016	*88-8039		
START	001000	69-581	#69-583							
STRTP	002120	#72-773	*72-808	90-8840	90-8843					
STRST	002116	#72-772	72-794	*72-807						
SWR	002064	*69-607	#70-633	72-786	72-790	72-798	82-4650	89-8711	90-8765	90-8827
		90-8830								
SWREG	000176	#68-557								
TDAR	= 000001	#71-736	77-2299	77-2339	77-2382	77-2429	77-2480	77-2530	79-2583	80-3596
		80-3703	80-3811	80-3816	81-4027	81-4134	81-4242	81-4247	81-4518	83-4987
		83-5112	83-5499	83-5553	84-5603	84-5661	84-5744	84-5829	84-5912	86-6771
		86-6787	86-6796	86-6909	86-6925	86-6934				
TITL	045320	90-8756	#90-8765							
TPB	= 001000	#71-742	83-5516	83-5570	84-5624	84-5701	84-5783	84-5869	84-5951	84-6117
TPE	- 000040	#71-751								
TSTCNT	002130	#72-777	*72-800	72-801	*72-805					
TSTID	- 000230	#71-756	74-870	74-870	#74-870	74-896	74-896	#74-896	74-922	74-922
		#74-922	74-948	74-948	#74-948	74-974	74-974	#74-974	74-1000	74-1000
		#74-1000	74-1022	74-1022	#74-1022	74-1049	74-1049	#74-1049	74-1069	74-1069
		#74-1069	74-1090	74-1090	#74-1090	74-1111	74-1111	#74-1111	74-1132	74-1132
		#74-1132	74-1154	74-1154	#74-1154	74-1173	74-1173	#74-1173	74-1194	74-1194
		#74-1194	74-1213	74-1213	#74-1213	75-1235	75-1235	#75-1235	75-1253	75-1253
		#75-1253	75-1271	75-1271	#75-1271	75-1296	75-1296	#75-1296	75-1317	75-1317
		#75-1317	75-1338	75-1338	#75-1338	75-1358	75-1358	#75-1358	75-1378	75-1378
		#75-1378	75-1396	75-1396	#75-1396	75-1416	75-1416	#75-1416	75-1443	75-1443
		#75-1443	75-1460	75-1460	#75-1460	75-1471	75-1471	#75-1471	75-1482	75-1482
		#75-1482	75-1493	75-1493	#75-1493	75-1504	75-1504	#75-1504	75-1515	75-1515
		#75-1515	75-1526	75-1526	#75-1526	75-1537	75-1537	#75-1537	75-1548	75-1548
		#75-1548	75-1559	75-1559	#75-1559	75-1570	75-1570	#75-1570	75-1590	75-1590
		#75-1590	75-1612	75-1612	#75-1612	75-1644	75-1644	#75-1644	75-1664	75-1664
		#75-1664	75-1684	75-1684	#75-1684	75-1704	75-1704	#75-1704	75-1724	75-1724
		#75-1724	75-1746	75-1746	#75-1746	75-1779	75-1779	#75-1779	75-1795	75-1795
		#75-1795	75-1806	75-1806	#75-1806	75-1832	75-1832	#75-1832	*76-1904	76-1904
		#76-1904	77-1930	77-1930	#77-1930	77-1985	77-1985	#77-1985	77-2024	77-2024
		#77-2024	77-2082	77-2082	#77-2082	77-2165	77-2165	#77-2165	77-2209	77-2209
		#77-2209	77-2291	77-2291	#77-2291	77-2331	77-2331	#77-2331	77-2371	77-2371
		#77-2371	77-2418	77-2418	#77-2418	77-2469	77-2469	#77-2469	77-2519	77-2519
		#77-2519	79-2572	79-2572	#79-2572	79-2634	79-2634	#79-2634	79-2680	79-2680
		#79-2680	79-2751	79-2751	#79-2751	79-2869	79-2869	#79-2869	79-2941	79-2941

SYMBOL	VALUE	REFERENCES
T112	015620	#79-3139
T113	016016	#79-3185
T114	016216	#80-3230
T115	016440	#80-3306
T116	016634	#80-3357
T117	017230	#80-3465
T12	003406	#74-1090
T120	017324	#80-3492
T121	017402	#80-3515
T122	017510	#80-3543
T123	017616	#80-3578
T124	020034	#80-3634
T125	020236	#80-3685
T126	020454	#80-3741
T127	020656	#80-3791
T13	003502	#74-1111
T130	021124	#80-3863
T131	021402	#81-3931
T132	021660	#81-4009
T133	022076	#81-4065
T134	022300	#81-4116
T135	022516	#81-4172
T136	022720	#81-4222
T137	023166	#81-4294
T14	003552	#74-1132
T140	023444	#81-4359
T141	023722	#81-4435
T142	024146	#81-4500
T143	024356	#82-4556
T144	024556	#82-4604
T145	025030	#82-4694
T146	025212	#82-4743
T147	025532	#82-4850
T15	003646	#74-1154
T150	026052	#83-4965
T151	026446	#83-5090
T152	027042	#83-5225
T153	027204	#83-5281
T154	027376	#83-5347
T155	027562	#83-5414
T156	027742	#83-5478
T157	030070	#83-5529
T16	003716	#74-1173
T160	030244	#83-5587
T161	030410	#84-5637
T162	030640	#84-5720
T163	031070	#84-5803
T164	031324	#84-5888
T165	031560	#84-5980
T166	032034	#84-6077
T167	032240	#84-6157
T17	003766	#74-1194

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
T170		032342	#84-6187
T171		032542	#85-6249
T172		033100	#85-6361
T173		033436	#86-6492
T174		033726	#86-6560
T175		034216	#86-6628
T176		034574	#86-6746
T177		035226	#86-6884
T2		002572	#74-896
T20		004036	#74-1213
T200		035660	#86-7033
T201		036214	#87-7146
T202		036550	#87-7267
T203		037104	#87-7379
T204		037440	#87-7508
T205		037562	#87-7562
T206		037704	#87-7616
T207		040026	#87-7672
T21		004116	#75-1235
T210		040212	#88-7736
T211		040376	#88-7804
T212		040520	#88-7860
T213		040650	#88-7913
T214		041016	#88-7983
T215		041230	#88-8058
T216		041346	#88-8116
T217		041464	#88-8184
T22		004162	#75-1253
T220		041746	#88-8270
T221		042114	#88-8330
T222		042644	#89-8432
T223		043240	#89-8522
T224		043404	#89-8566
T225		043552	#89-8618
T226		043706	#89-8658
T227		044010	#89-8699
T23		004232	#75-1271
T24		004302	#75-1296
T25		004352	#75-1317
T26		004422	#75-1338
T27		004472	#75-1358
T3		002666	#74-922
T30		004542	#75-1378
T31		004612	#75-1396
T32		004662	#75-1416
T33		004732	#75-1443
T34		005002	#75-1460
T35		005052	#75-1471
T36		005122	#75-1482
T37		005172	#75-1493
T4		002762	#74-948
T40		005242	#75-1504

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES										
T41		005312	#75-1515										
T42		005362	#75-1526										
T43		005432	#75-1537										
T44		005502	#75-1548										
T45		005552	#75-1559										
T46		005622	#75-1570										
T47		005672	#75-1590										
T5		003056	#74-974										
T50		005750	#75-1612										
T51		006026	#75-1644										
T52		006104	#75-1664										
T53		006162	#75-1684										
T54		006240	#75-1704										
T55		006316	#75-1724										
T56		006406	#75-1746										
T57		006504	#75-1779										
T6		003152	#74-1000										
T60		006554	#75-1795										
T61		006624	#75-1806										
T62		006674	#75-1832										
T63		007172	#76-1904										
T64		007260	#77-1930										
T65		007456	#77-1985										
T66		007636	#77-2024										
T67		010102	#77-2082										
T7		003222	#74-1022										
T70		010460	#77-2165										
T71		010626	#77-2209										
T72		011156	#77-2291										
T73		011344	#77-2331										
T74		011536	#77-2371										
T75		011746	#77-2418										
T76		012166	#77-2469										
T77		012412	#77-2519										
UCB	-	001000	#71-730	75-1260	82-4636	82-4715	87-7074	87-7079	87-7186	87-7191	87-7307		
			87-7312	87-7419	87-7424	88-7893	88-7954	88-7958					
UMPROC	-	170200	#71-702	*76-1891									
UMPR01	=	170202	#71-703	*76-1892									
UMPR02	=	170204	#71-704	*76-1893									
UMPR03	-	170206	#71-705	*76-1894									
UMPR04	-	170210	#71-706	*76-1895									
UMPR05	-	170212	#71-707	*76-1896									
UMPR06	=	170214	#71-708	*76-1897									
UMPR07	-	170216	#71-709	*76-1898									
UMPR08	-	170220	#71-710	*77-2095	*77-2157								
UMPR09	-	170222	#71-711	*77-2106									
VCIP		010000	#71-732	80-3474	80-3500	80-3525	80-3528	80-3553	80-3556	80-3588	80-3644		
			80-3695	80-3751	80-3800	81-3877	81-3892	81-3895	81-3945	81-3960	81-3963		
			81-4019	81-4075	81-4126	81-4182	81-4231	81-4307	81-4322	81-4325	81-4372		
			81-4387	81-4390	82-4625	82-4701	83-5448	84-5622	84-6204	85-6281	85-6318		
			85-6396	85-6436	86-6505	86-6545	86-6573	86-6612	86-6649	86-6701	87-7054		
			87-7166	87-7287	87-7399	88-7718	88-7786	88-8340	89-8442				

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES								
VLD	-	010000	#71-745	80-3612	80-3662	80-3719	80-3769	80-3828	81-3904	81-3972	81-4043
			81-4093	81-4150	81-4200	91-4259	81-4334	81-4399	82-4654	82-4662	82-4671
			82-4679	82-4725	89-8406						
VSIU	=	020000	#71-733	80-3522	80-3531	80-3550	80-3559	80-3585	80-3641	80-3692	80-3748
			80-3797	81-3874	81-3942	81-4016	81-4072	81-4123	81-4179	81-4228	81-4304
			81-4369	82-4622	82-4698	87-7051	87-7163	87-7284	87-7396	88-8337	89-8439
WWPD	-	000100	#71-727	83-5434	85-6406	85-6412	86-6519	86-6523	86-6587	86-6591	86-6665
			86-6669	88-7765	88-7771						
WWPT	=	002000	#71-731	84-5609	85-6290	85-6295	87-7699	88-7705			
\$APTHD		001444	70-618	#70-618							
\$ASTAT	=	*****	70-620	70-620							
\$ATYC		001630	70-620	#70-620							
\$ATY1		001604	#70-620								
\$ATY3		001612	#70-620	90-8740							
\$ATY4		001622	#70-620								
\$BASE		001534	#70-619								
\$BELL		002102	#70-642								
\$CDW1		001540	#70-619								
\$CDW2		001542	#70-619								
\$CHARC		044522	*90-8740	*90-8740	90-8740	*90-8740	#90-8740				
\$CKSWR	=	*****	90-8743								
\$CPUOP		001506	#70-619								
\$CRLF		002107	#70-644	90-8740	90-8740	90-8740					
\$DBLK		045170	90-8742	90-8742	#90-8742						
\$DDW0		001544	#70-619								
\$DDW1		001546	#70-619								
\$DDW10		001570	#70-619								
\$DDW11		001572	#70-619								
\$DDW12		001574	#70-619								
\$DDW13		001576	#70-619								
\$DDW14		001600	#70-619								
\$DDW15		001602	#70-619								
\$DDW2		001550	#70-619								
\$DDW3		001552	#70-619								
\$DDW4		001554	#70-619								
\$DDW5		001556	#70-619								
\$DDW6		001560	#70-619								
\$DDW7		001562	#70-619								
\$DDW8		001564	#70-619								
\$DDW9		001566	#70-619								
\$DEVCT		001470	#70-619								
\$DEVVM		001536	#70-619								
\$DOAGN		044164	89-8716	89-8724	#89-8730						
\$DTBL		045160	90-8742	#90-8742							
\$ENDAD		044154	68-576	#89-8726							
\$ENULL		002112	#70-646	89-8722	90-8790	90-8803					
\$ENV		001500	#70-619	70-620	70-620	90-8740	90-8823				
\$ENVM		001501	69-605	#70-619	70-620	79-2764	88-8345	89-8447	90-8740	90-8740	
\$ERROR		045252	68-569	#90-8750							
\$ETABL		001500	#70-619								
\$ETEND		001604	70-618	#70-619							
\$FATAL		001462	#70-619	*70-620							

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
\$FFLG		002050	*70-620 *70-620 70-620 *70-620 #70-620
\$FILLC		002100	#70-640 90-8740 90-8740 90-8740
\$FILLS		002077	#70-639 90-8740 90-8740
\$GTSWR	=	*****	90-8743
\$HD	=	000003	68-555 68-555 68-555
\$HIBTS		001444	#70-618
\$ICNT		044172	*89-8714 89-8715 *89-8717 #89-8733
\$LF		002110	#70-645 90-8740 90-8740
\$LFLG		002047	*70-620 #70-620
\$MADR1		001512	#70-619
\$MADR2		001516	#70-619
\$MADR3		001522	#70-619
\$MADR4		001526	#70-619
\$MAIL		001460	70-618 70-618 #70-619 90-8740
\$MAMS1		001510	#70-619
\$MAMS2		001514	#70-619
\$MAMS3		001520	#70-619
\$MAMS4		001524	#70-619
\$MBADR		001446	#70-618
\$MFLG		002046	*70-620 70-620 *70-620 #70-620
\$MSGAD		001474	#70-619 *70-620 70-620
\$MSGLG		001476	#70-619 *70-620
\$MSGTY		001460	#70-619 70-620 *70-620 70-620 *70-620
\$MTYP1		001511	#70-619
\$MTYP2		001515	#70-619
\$MTYP3		001521	#70-619
\$MTYP4		001525	#70-619
\$NULL		002076	#70-638 90-8740 90-8740 90-8740
\$OCNT		044750	*90-8741 *90-8741 #90-8741
\$OMODE		044752	*90-8741 *90-8741 90-8741 *90-8741 *90-8741 #90-8741
\$PASS		001466	*69-604 #70-619 72-796 79-2842 89-8363 #90-8741
\$PASTM		001452	#70-618
\$QUES		002106	#70-643 90-8740 90-8740
\$RDCHR	=	*****	90-8743
\$RDDEC	-	*****	90-8743
\$RDLIN	-	*****	90-8743
\$RDOCT	-	*****	90-8743
\$R2A	=	*****	90-8743
\$SAVRE	-	*****	90-8743
\$SCPSE		002134	68-566 #72-782
\$SWR	-	160000	68-555 #68-555
\$SWREG		001502	69-607 #70-619
\$TESTN		001464	#70-619 72-784 72-788 *72-793 *73-851 *74-870 *74-896 *74-922 *74-948
			*74-974 *74-1000 *74-1022 *74-1049 *74-1069 *74-1090 *74-1111 *74-1132 *74-1154
			*74-1173 *74-1194 *74-1213 *75-1235 *75-1253 *75-1271 *75-1296 *75-1317 *75-1338
			*75-1358 *75-1378 *75-1396 *75-1416 *75-1443 *75-1460 *75-1471 *75-1482 *75-1493
			*75-1504 *75-1515 *75-1526 *75-1537 *75-1548 *75-1559 *75-1570 *75-1590 *75-1612
			*75-1644 *75-1664 *75-1684 *75-1704 *75-1724 *75-1746 *75-1779 *75-1795 *75-1806
			*75-1832 *76-1904 *77-1930 *77-1985 *77-2024 *77-2082 *77-2165 *77-2209 *77-2291
			*77-2331 *77-2371 *77-2418 *77-2469 *77-2519 *77-2572 *77-2634 *77-2680 *77-2751
			*79-2869 *79-2941 *79-2976 *79-3011 *79-3051 *79-3095 *79-3139 *79-3185 *80-3230
			*80-3306 *80-3357 *80-3465 *80-3492 *80-3515 *80-3543 *80-3578 *80-3634 *80-3685

SYMBOL CROSS REFERENCE
SYMBOL VALUE

REF

REFERENCES

SYMBOL	VALUE	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES
		*80-3741	*80-3791	*80-3863	*81-3931	*81-4009	*81-4065	*81-4116	*81-4172	*81-4222
		*81-4294	*81-4359	*81-4435	*81-4500	*82-4556	*82-4604	*82-4694	*82-4743	*82-4850
		*83-4965	*83-5090	*83-5225	*83-5281	*83-5347	*83-5414	*83-5478	*83-5529	*83-5587
		*84-5637	*84-5720	*84-5803	*84-5888	*84-5980	*84-6077	*84-6157	*84-6187	*85-6249
		*85-6361	*86-6492	*86-6560	*86-6628	*86-6746	*86-6884	*86-7033	*87-7146	*87-7267
		*87-7379	*87-7508	*87-7562	*87-7616	*87-7672	*88-7736	*88-7804	*88-7860	*88-7913
		*88-7983	*88-8058	*88-8116	*88-8184	*88-8270	*88-8330	*89-8432	*89-8522	*89-8566
		*89-8618	*89-8658	*89-8699	90-8787					
		89-8715	#89-8732							
\$TIMES	044170	#70-635	90-8740	90-8740						
\$TKB	002070	#70-634	90-8740	90-8740						
\$TKS	002066	68-555	#68-555							
\$TN	= 000001	#70-637	90-8740	90-8740	90-8740					
\$TPB	002074	#70-641	90-8740	90-8740	90-8740					
\$TPFLG	002101	#70-636	90-8740	90-8740	90-8740					
\$TPS	002072	68-571	#90-8743							
\$TRAP	045200	#90-8743	90-8743							
\$TRAP2	045222	#90-8743	90-8743	90-8743	90-8743	90-8743	#90-8743	90-8743	90-8743	90-8743
\$TRP	- 000006	90-8743	#90-8743	90-8743	90-8743	90-8743	90-8743	#90-8743	90-8743	90-8743
		90-8743	90-8743	#90-8743	90-8743	90-8743	90-8743	90-8743	#90-8743	
		90-8743	#90-8743							
\$TRPAD	045234	90-8743	#90-8743							
\$STSM	001450	#70-618								
\$STPBN	*****	90-8743								
\$STPDS	044754	#90-8742	90-8743	90-8743						
\$STYPE	044174	70-620	#90-8740	90-8743	90-8743					
\$STYPEC	044406	90-8740	90-8740	90-8740	#90-8740	90-8740				
\$STYPEX	044524	90-8740	90-8740	#90-8740						
\$STYPOC	044552	#90-8741	90-8743	90-8743						
\$STYPON	044566	90-8741	#90-8741	90-8743						
\$STYPOS	044526	#90-8741	90-8743							
\$UNIT	001472	#70-619								
\$UNITM	001454	#70-618								
\$USWR	001504	#70-619	79-2766	88-8347	89-8449					
\$VECT1	001530	#70-619								
\$VECT2	001532	#70-619								
\$OF ILL	044751	*90-8741	*90-8741	90-8741	#90-8741					
.\$ASTA	*****	70-620	70-620							
.\$X	001444	#70-618	70-618							

MACRO NAME	REFERENCES									
COMMEN	#18-1530									
ENDCOM	#18-1542									
EOTST	#67-473	75-1459	75-1470	75-1481	75-1492	75-1503	75-1514	75-1525	75-1536	75-1547
	75-1558	75-1569	75-1794	75-1805						
ERR	#67-478	74-887	74-913	74-939	74-965	74-991	74-1013	74-1034	74-1059	74-1082
	74-1103	74-1122	74-1145	74-1165	74-1184	74-1205	74-1226	75-1264	75-1281	75-1307
	75-1329	75-1349	75-1369	75-1389	75-1407	75-1427	75-1455	75-1466	75-1477	75-1488
	75-1499	75-1510	75-1521	75-1532	75-1543	75-1554	75-1565	75-1576	75-1603	75-1624
	75-1656	75-1676	75-1696	75-1716	75-1738	75-1761	75-1790	75-1801	75-1812	75-1862
	77-1921	77-1971	77-2010	77-2067	77-2147	77-2195	77-2263	77-2322	77-2362	77-2403
	77-2453	77-2504	77-2554	79-2607	79-2665	79-2731	79-2823	79-2912	79-2966	79-3001
	79-3038	79-3081	79-3124	79-3169	79-3214	80-3275	80-3345	80-3430	80-3482	80-3505
	80-3534	80-3562	80-3618	80-3668	80-3725	80-3775	80-3834	81-3912	81-3980	81-4049
	81-4099	81-4156	81-4206	81-4265	81-4341	81-4407	81-4474	81-4486	82-4540	82-4590
	82-4656	82-4664	82-4673	82-4681	82-4727	82-4800	82-4826	82-4907	82-4933	83-5029
	83-5063	83-5154	83-5188	83-5263	83-5272	83-5321	83-5332	83-5388	83-5400	83-5453
	83-5463	83-5519	83-5572	84-5626	84-5681	84-5693	84-5703	84-5764	84-5775	84-5785
	84-5849	84-5861	84-5871	84-5932	84-5943	84-5953	84-6033	84-6053	84-6122	84-6176
	84-6235	85-6325	85-6337	85-6349	85-6443	85-6455	85-6467	86-6549	86-6616	86-6705
	86-6713	86-6721	86-6728	86-6828	86-6861	86-6966	86-6999	87-7103	87-7125	87-7215
	87-7237	87-7336	87-7358	87-7448	87-7470	87-7551	87-7605	87-7658	88-7723	88-7791
	88-7845	88-7902	88-7970	88-8046	88-8103	88-8163	88-8246	88-8257	88-8300	89-8412
	89-8496	89-8505	89-8554	89-8598	89-8645	89-8690				
ESCAPE	#20-1658									
GETPRI	#13-1282									
GETSWR	#22-1729									
LNOP	#67-466	74-882	74-908	74-934	74-960	74-986	74-1010	74-1032	74-1057	74-1079
	74-1098	74-1119	74-1141	74-1162	74-1181	74-1202	74-1222	75-1243	75-1261	75-1278
	75-1304	75-1326	75-1346	75-1366	75-1386	75-1404	75-1424	75-1452	75-1463	75-1474
	75-1485	75-1496	75-1507	75-1518	75-1529	75-1540	75-1551	75-1562	75-1573	75-1600
	75-1621	75-1652	75-1672	75-1692	75-1712	75-1734	75-1757	75-1787	75-1798	75-1809
	75-1859	77-1918	77-1958	77-2002	77-2059	77-2132	77-2187	77-2245	77-2311	77-2351
	77-2393	77-2440	77-2491	77-2541	79-2594	79-2654	79-2713	79-2805	79-2904	79-2961
	79-2996	79-3032	79-3074	79-3117	79-3162	79-3207	80-3267	80-3331	80-3410	80-3479
	80-3504	80-3530	80-3558	80-3609	80-3659	80-3716	80-3766	80-3825	81-3897	81-3965
	81-4040	81-4090	81-4147	81-4197	81-4256	81-4327	81-4392	81-4466	82-4529	82-4578
	82-4648	82-4723	82-4790	82-4897	83-5012	83-5137	83-5258	83-5316	83-5382	83-5443
	83-5513	83-5567	84-5618	84-5674	84-5757	84-5842	84-5925	84-6024	84-6114	84-6171
	84-6225	85-6312	85-6430	86-6538	86-6605	86-6694	86-6809	86-6947	87-7093	87-7205
	87-7326	87-7438	87-7546	87-7600	87-7653	88-7715	88-7783	88-7840	88-7898	88-7965
	88-8037	88-8098	88-8158	88-8230	88-8292	89-8403	89-8491	89-8545	89-8589	89-8654
	89-8685	89-8702								
MULT	#43-4409									
NEWST	#19-1589									
POP	#26-2107	#67-449	70-620	70-620	90-8742					
PUSH	#26-2099	#67-449	70-620	70-620	70-620	90-8742				
REPORT	#56-5368									
SCPSET	#67-487	#74-876	#74-902	#74-928	#74-954	#74-980	#74-1008	#74-1030	#74-1055	#74-1075
	#74-1096	#74-1117	#74-1138	#74-1160	#74-1179	#74-1200	#74-1219	#75-1241	#75-1259	#75-1276
	#75-1302	#75-1324	#75-1344	#75-1364	#75-1384	#75-1402	#75-1422	#75-1450	#75-1461	#75-1472
	#75-1483	#75-1494	#75-1505	#75-1516	#75-1527	#75-1538	#75-1549	#75-1560	#75-1571	#75-1597
	#75-1618	#75-1649	#75-1669	#75-1689	#75-1709	#75-1729	#75-1751	#75-1785	#75-1796	#75-1807

MACRO CROSS REFERENCE

REF

MACRO NAME

REFERENCES

SETPRI
SETTRA
SETUP
SKIP
SLASH
STARS

#75-1840	#77-1912	#77-1941	#77-1992	#77-2032	#77-2089	#77-2174	#77-2218	#77-2297	#77-2337
#77-2377	#77-2424	#77-2475	#77-2525	#79-2578	#79-2642	#79-2688	#79-2763	#79-2883	#79-2948
#79-2983	#79-3018	#79-3059	#79-3102	#79-3147	#79-3192	#80-3241	#80-3318	#80-3381	#80-3470
#80-3497	#80-3521	#80-3549	#80-3584	#80-3640	#80-3691	#80-3747	#80-3796	#80-3870	#81-3939
#81-4015	#81-4071	#81-4122	#81-4178	#81-4227	#81-4301	#81-4366	#81-4442	#81-4509	#82-4560
#82-4621	#82-4697	#82-4759	#82-4866	#83-4981	#83-5106	#83-5244	#83-5300	#83-5366	#83-5428
#83-5497	#83-5549	#84-5601	#84-5653	#84-5736	#84-5819	#84-5904	#84-5995	#84-6092	#84-6163
#84-6202	#85-6279	#85-6394	#86-6503	#86-6571	#86-6647	#86-6762	#86-6900	#87-7050	#87-7162
#87-7283	#87-7395	#87-7532	#87-7585	#87-7639	#87-7691	#88-7756	#88-7827	#88-7883	#88-7941
#88-8007	#88-8082	#88-8141	#88-8191	#88-8275	#88-8336	#89-8438	#89-8528	#89-8572	#89-8626
#89-8665	#89-8700								
#12-1250									
#90-8743	#90-8743	#90-8743	#90-8743	#90-8743	#90-8743				
#14-1306									
#21-1692									
#17-1482									
#16-1451	#67-447	#67-448	70-618	70-618	70-618	70-619	70-620	70-622	70-624
72-763	72-769	72-819	72-821	74-870	74-871	74-875	74-896	74-897	74-901
74-922	74-923	74-927	74-948	74-949	74-953	74-974	74-975	74-979	74-1000
74-1001	74-1007	74-1022	74-1023	74-1029	74-1049	74-1050	74-1054	74-1069	74-1070
74-1074	74-1090	74-1091	74-1095	74-1111	74-1112	74-1116	74-1132	74-1133	74-1137
74-1154	74-1155	74-1159	74-1173	74-1174	74-1178	74-1194	74-1195	74-1199	74-1213
74-1214	74-1218	75-1235	75-1236	75-1240	75-1253	75-1254	75-1258	75-1271	75-1272
75-1275	75-1296	75-1297	75-1301	75-1317	75-1318	75-1322	75-1338	75-1339	75-1343
75-1358	75-1359	75-1363	75-1378	75-1379	75-1383	75-1396	75-1397	75-1401	75-1416
75-1417	75-1421	75-1443	75-1444	75-1448	75-1460	75-1471	75-1482	75-1493	75-1504
75-1515	75-1526	75-1537	75-1548	75-1559	75-1570	75-1590	75-1591	75-1596	75-1612
75-1613	75-1617	75-1644	75-1645	75-1648	75-1664	75-1665	75-1668	75-1684	75-1685
75-1688	75-1704	75-1705	75-1708	75-1724	75-1725	75-1728	75-1746	75-1747	75-1750
75-1779	75-1780	75-1784	75-1795	75-1806	75-1832	75-1839	76-1904	76-1910	77-1930
77-1931	77-1940	77-1985	77-1991	77-2024	77-2031	77-2082	77-2088	77-2165	77-2173
77-2209	77-2217	77-2291	77-2292	77-2296	77-2331	77-2332	77-2336	77-2371	77-2372
77-2376	77-2418	77-2423	77-2469	77-2474	77-2519	77-2524	79-2572	79-2577	79-2634
79-2641	79-2680	79-2687	79-2751	79-2762	79-2869	79-2882	79-2941	79-2942	79-2947
79-2976	79-2977	79-2982	79-3011	79-3012	79-3017	79-3051	79-3052	79-3057	79-3095
79-3096	79-3101	79-3139	79-3140	79-3145	79-3185	79-3186	79-3191	80-3230	80-3233
80-3240	80-3306	80-3307	80-3317	80-3357	80-3358	80-3380	80-3465	80-3469	80-3492
80-3496	80-3515	80-3520	80-3543	80-3548	80-3578	80-3583	80-3634	80-3639	80-3685
80-3690	80-3741	80-3746	80-3791	80-3795	80-3863	80-3868	81-3931	81-3936	81-4009
81-4014	81-4065	81-4070	81-4116	81-4121	81-4172	81-4177	81-4222	81-4226	81-4294
81-4299	81-4359	81-4364	81-4435	81-4441	81-4500	81-4508	82-4556	82-4559	82-4604
82-4620	82-4694	82-4696	82-4743	82-4758	82-4850	82-4865	83-4965	83-4980	83-5090
83-5105	83-5225	83-5243	83-5281	83-5299	83-5347	83-5365	83-5414	83-5427	83-5478
83-5496	83-5529	83-5548	83-5587	84-5600	84-5637	84-5652	84-5720	84-5735	84-5803
84-5818	84-5888	84-5903	84-5980	84-5994	84-6077	84-6091	84-6157	84-6162	84-6187
84-6200	85-6249	85-6277	85-6361	85-6392	86-6492	86-6502	86-6560	86-6570	86-6628
86-6646	86-6746	86-6761	86-6884	86-6899	86-7033	87-7049	87-7146	87-7161	87-7267
87-7282	87-7379	87-7394	87-7508	87-7531	87-7562	87-7584	87-7616	87-7638	87-7672
87-7690	88-7736	88-7755	88-7804	88-7826	88-7860	88-7882	88-7913	88-7940	88-7983
88-8006	88-8058	88-8081	88-8116	88-8140	88-8184	88-8190	88-8270	88-8274	88-8330
88-8335	89-8432	89-8437	89-8522	89-8527	89-8566	89-8571	89-8618	89-8625	89-8658
89-8459	89-8664	89-8699	89-8708	90-8740	90-8741	90-8742	90-8743	90-8745	90-9005

MACRO NAME	REFERENCES
.\$POWE	#40-4159
.\$RAND	#41-4234
.\$RDDE	#37-3830 #67-448
.\$RDOC	#36-3739 #67-448
.\$READ	#35-3344 #67-451
.\$R2AZ	#51-4874
.\$SAVE	#38-3905
.\$SB2D	#47-4691
.\$SB2O	#49-4792
.\$SCOP	#28-2401
.\$SIZE	#42-4287
.\$SUPR	#50-4830
.\$STRAP	#39-4007 #67-453 #90-8743
.\$TYPB	#34-3237
.\$TYPD	#33-3160 #67-451 #90-8742
.\$TYPE	#31-2929 #67-451 90-8740
.\$TYPO	#32-3064 #67-450 #90-8741
.\$4OCA	#8-948
.1170	#6-502