

.REM %

IDENTIFICATION

PRODUCT CODE:	AC-F624C-MC
PRODUCT NAME:	CKKKACO 11/44 KK11B CACHE
DATE CREATED:	APRIL, 1981
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	DAN P. MILLEVILLE

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, 1981 BY DIGITAL EQUIPMENT CORPORATION

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

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80

TABLE OF CONTENTS

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

81
82
83
84
85
86
87
88
89
90
91
92
93

1.0 HISTORY SECTION

- CKKKAAG WAS RELEASED OCT 1979
- CKKKABO WAS RELEASED OCT 1980
- CKKKACO WAS RELEASED APR 1981
- POWER MONITOR CHECK BEFORE EACH TEST & ON ERROR.
- NEW SYSMAC VERSION C5 TO CLEAN UP XON-XOFF PROBLEMS.
- SET ERROR INDICATOR (\$MSGTYP) ON ERROR FOR APT.
- FIXED 10W TEMP PROBLEM WITH TEST 225 - CLEAR CME REGISTER BEFORE EACH TEST.

94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138

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.

139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193

2.2 TEST STRUCTURE

EACH TEST IS STRUCTURED WITH THE FOLLOWING DEDICATED LOCAL SYMBOLS:

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

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

2.3 HARDWARE REQUIREMENTS

2.3.1 REQUIRED EQUIPMENT

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

2.3.2 OPTIONAL EQUIPMENT

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

2.3.3 DIAGNOSTIC PREREQUISITES

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

2.3.4 RELATED DOCUMENTS

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

194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243

3.0 OPERATING INSTRUCTIONS

3.1 LOAD AND START PROCEDURE

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

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

3.2 SWITCH REGISTER OPTIONS

3.2.1 [OPTIONS]

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

244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291

3.2.2 LOOP ON ERROR

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

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

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

3.2.3 LOOP ON TEST

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

3.2.4 IMPLEMENTATION

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

292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344

3.3 APT

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

BIT 12 \$USWR (UNIBUS EXERCISER)

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

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

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

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

1ST PASS	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

345
346
347
348
349
350

3.4 EXECUTION TIMES

1ST PASS:
PASSES WITH ITERATIONS:

LESS THAN 10 SEC.
LESS THAN 75 SEC.

351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394

4.0 ERROR INFO

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

<u>TESTNO</u>	<u>ERRPC</u>	<u>CHR157</u>	<u>CA2113</u>
102	13234	001	000

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

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

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

4.2 UNCONTROLLED ERRORS

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

4.3 POWER MONITOR BIT ERRORS

IF THE POWER MONITOR BIT IS FOUND SET IN THE SCOPE ROUTINE, AN ERROR WILL CALL FROM THE SCOPE ROUTINE. LOOP-ON-ERROR IS DISABLED FOR THIS ERROR ONLY. IF THE POWER MONITOR BIT BECOMES SET AFTER THE SCOPE AND FOR ANY REASON A FAILURE OCCURS IN THAT TEST, THE ERROR CALL WILL CALL *TWO* ERRORS, THE FIRST ERROR BEING THE POWER MONITOR BIT ERROR, THEN THE ERROR ORIGINALLY TO BE CALLED.

395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426

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).%

517

```
.TITLE CKKKACO 11-44 KK11B CACHE
;*COPYRIGHT (C) APR 1981
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY DAN P. MILLEVILLE
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
;*
```

000001
160000

518

```
$TN=1
$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
.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 000000
000176 000000

```
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
```

000200 000137 000200
525 000020 000020
526 000020 002144
527 000022 000340
528 000030 000030
529 000030 046224
530 000032 000340
531 000034 046116
532 000036 000340
533 000042 000042
534 000042 000000
535 000046 000046
536 000046 044222
537 000052 000052
538 000052 000000
539 000004 000004

```
.SBTTL STARTING ADDRESS(ES)
JMP @#200 ;;JUMP TO STARTING ADDRESS OF PROGRAM
.=20
.WORD $SCPSET
.WORD 340
.-30
.WORD $ERROR
.WORD 340
.WORD $TRAP
.WORD 340
.=42
.WORD 0
.=46
.WORD $ENDAD
.-52
.WORD 0
```

SCOPE-4

```

540      000200 000200      .-200
541 000200 000137 001000  JMP      START
542      001000 001000      .-1000
543 001000 000005      START:  RESET      ;DISABLE ALL INTERRUPTS
544 001002 012706 000500      MOV      #500,SP ;SET STACK POINTER
545 001006 012737 001032 000004  MOV      #4$,4   ;SETUP FOR POSSIBLE NEX MEMORY TRAP
546 001014 012737 000340 000006  MOV      #340,6
547 001022 012737 001015 177746  MOV      #OFF,CCR ;DISABLE CACHE
548 001030 000536      BR      5$      ;NO TRAP;CONTINUE
549 001032 022626      4$:  CMP      (SP)+,(SP)+ ;ADJUST STACK DUE TO TRAP
550 001034 012737 000006 000004  MOV      #6,4   ;RESTORE TRAP VECTORS
551 001042 005037 000006      CLR      6
552
553 001046 104401 001054      TYPE     ,65$    ;;TYPE ASCIZ STRING
      001052 000416      BR      64$    ;;GET OVER THE ASCIZ
      ;;65$: .ASCIZ <CRLF>/CKKACO 11-44 KK11B CACHE/
554 001110      64$:
      001110 104401 001116      TYPE     ,67$    ;;TYPE ASCIZ STRING
      001114 000423      BR      66$    ;;GET OVER THE ASCIZ
      ;;67$: .ASCIZ <CRLF>/TRAP THRU NEX MEMORY VECTOR OCCURED/
555 001164      66$:
      001164 104401 001172      TYPE     ,69$    ;;TYPE ASCIZ STRING
      001170 000424      BR      68$    ;;GET OVER THE ASCIZ
      ;;69$: .ASCIZ <CRLF>/DIAGNOSTIC ATTEMPTED TO TURN CACHE OFF/
556 001242      68$:
      001242 104401 001250      TYPE     ,71$    ;;TYPE ASCIZ STRING
      001246 000423      BR      70$    ;;GET OVER THE ASCIZ
      ;;71$: .ASCIZ <CRLF>/BY ADDRESSING CACHE CONTROL REGISTER/
557 001316      70$:
      001316 012737 000001 001466  MOV      #1,$MSGTY ;SET $MSGTY FOR POSSIBLE APT USE
558 001324 000000      HALT      ;HALT PROGRAM
559
560 001326 012737 000006 000004  5$:  MOV      #6,4   ;RESTORE VECTORS
561 001334 005037 000006      CLR      6
562 001340 005037 001474      CLR     $PASS   ;CLEAR PASS COUNT
563 001344 132737 000200 001507  BITB     #APTSIZE,$ENVM ;IS APT SIZING?
564 001352 001403      BEQ     1$      ;NO
565 001354 012737 001510 002074  MOV     #$$SWREG,$SWR ;YES;USE APT SWITCH REGISTER
566 001362 005737 000042      1$:  TST     42      ;IS THIS MANUAL MODE?
567 001366 001404      BEQ     3$      ;YES TYPE ID
568 001370 023737 000042 000046  CMP     42,46   ;IS THIS ACT 11 QV OR AUTO MODE?
569 001376 001423      BEQ     2$      ;YES;SKIP TITLE
570 001400 000240      3$:  NOP
571 001402 000240      NOP
572 001404 104401 001412      TYPE     ,73$    ;;TYPE ASCIZ STRING
      001410 000416      BR      72$    ;;GET OVER THE ASCIZ
      ;;73$: .ASCIZ <CRLF>/CKKACO 11-44 KK11B CACHE /
573 001446      72$:
574 001446 000137 002500      2$:  JMP     BEGIN   ;START TEST

```

575

001452
000024
000024 000200
000044 000044
001452
001452

001452
001452 000000
001454 001466
001456 000005
001460 000010
001462 00000C
001464 000052

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

577

```
.SBTTL APT MAILBOX-ETABLE
*****
.EVEN
001466      000000  $MAIL:          ;;APT MAILBOX
001466      000000  $MSGTY: .WORD   AMSGTY ;;MESSAGE TYPE CODE
001470      000000  $FATAL: .WORD   AFATAL ;;FATAL ERROR NUMBER
001472      000000  $TESTN: .WORD   ATESTN ;;TEST NUMBER
001474      000000  $PASS:   .WORD   APASS  ;;PASS COUNT
001476      000000  $DEVCT: .WORD   ADEVCT ;;DEVICE COUNT
001500      000000  $UNIT:  .WORD   AUNIT  ;;I/O UNIT NUMBER
001502      000000  $MSGAD: .WORD   AMSGAD ;;MESSAGE ADDRESS
001504      000000  $MSGLG: .WORD   AMSGLG ;;MESSAGE LENGTH
001506      000000  $ETABLE:      ;;APT ENVIRONMENT TABLE
001506      000     $ENV:   .BYTE   AENV   ;;ENVIRONMENT BYTE
001507      000     $ENVM:  .BYTE   AENVM  ;;ENVIRONMENT MODE BITS
001510      000000  $$SWREG: .WORD   ASWREG ;;APT SWITCH REGISTER
001512      000000  $USWR:  .WORD   AUSWR  ;;USER SWITCHES
001514      000000  $CPUOP: .WORD   ACPUOP ;;CPU TYPE,OPTIONS
                        ;;BITS 15-11=CPU TYPE
                        ;;11/04-01,11/05=02,11/20-03,11/40-04,11/45=05
                        ;;11/70=06,PDQ=07,Q-10
                        ;;BIT 10=REAL TIME CLOCK
                        ;;BIT 9=FLOATING POINT PROCESSOR
                        ;;BIT 8=MEMORY MANAGEMENT
001516      000     $MAMS1: .BYTE   AMAMS1 ;;HIGH ADDRESS,M.S. BYTE
001517      000     $MTYP1: .BYTE   AMTYP1 ;;MEM. TYPE,BLK#1
                        ;;MEM.TYPE BYTE -- (HIGH BYTE)
                        ;;900 NSEC CORE=001
                        ;;300 NSEC BIPOLAR=002
                        ;;500 NSEC MOS=003
001520      000000  $MADR1: .WORD   AMADR1 ;;HIGH ADDRESS,BLK#1
                        ;;MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
001522      000     $MAMS2: .BYTE   AMAMS2 ;;HIGH ADDRESS,M.S. BYTE
001523      000     $MTYP2: .BYTE   AMTYP2 ;;MEM. TYPE,BLK#2
001524      000000  $MADR2: .WORD   AMADR2 ;;MEM.LAST ADDRESS,BLK#2
001526      000     $MAMS3: .BYTE   AMAMS3 ;;HIGH ADDRESS,M.S.BYTE
001527      000     $MTYP3: .BYTE   AMTYP3 ;;MEM. TYPE,BLK#3
001530      000000  $MADR3: .WORD   AMADR3 ;;MEM.LAST ADDRESS,BLK#3
001532      000     $MAMS4: .BYTE   AMAMS4 ;;HIGH ADDRESS,M.S.BYTE
001533      000     $MTYP4: .BYTE   AMTYP4 ;;MEM. TYPE,BLK#4
001534      000000  $MADR4: .WORD   AMADR4 ;;MEM.LAST ADDRESS,BLK#4
001536      000000  $VECT1: .WORD   AVECT1 ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001540      000000  $VECT2: .WORD   AVECT2 ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001542      000000  $BASE:  .WORD   ABASE  ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001544      000000  $DEVN:  .WORD   ADEVN  ;;DEVICE MAP
001546      000000  $CDW1:  .WORD   ACDW1  ;;CONTROLLER DESCRIPTION WORD#1
001550      000000  $CDW2:  .WORD   ACDW2  ;;CONTROLLER DESCRIPTION WORD#2
001552      000000  $DDW0:  .WORD   ADDW0  ;;DEVICE DESCRIPTOR WORD#0
001554      000000  $DDW1:  .WORD   ADDW1  ;;DEVICE DESCRIPTOR WORD#1
001556      000000  $DDW2:  .WORD   ADDW2  ;;DEVICE DESCRIPTOR WORD#2
001560      000000  $DDW3:  .WORD   ADDW3  ;;DEVICE DESCRIPTOR WORD#3
001562      000000  $DDW4:  .WORD   ADDW4  ;;DEVICE DESCRIPTOR WORD#4
001564      000000  $DDW5:  .WORD   ADDW5  ;;DEVICE DESCRIPTOR WORD#5
001566      000000  $DDW6:  .WORD   ADDW6  ;;DEVICE DESCRIPTOR WORD#6
001570      000000  $DDW7:  .WORD   ADDW7  ;;DEVICE DESCRIPTOR WORD#7
001572      000000  $DDW8:  .WORD   ADDW8  ;;DEVICE DESCRIPTOR WORD#8
001574      000000  $DDW9:  .WORD   ADDW9  ;;DEVICE DESCRIPTOR WORD#9
```

CKKKACO 11-44 KK119 CACHE
APT MAILBOX-ETABLE

MATRO M1113 28-MAR-81 14:20 PAGE 16-1

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

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

579

```

.SBTTL APT COMMUNICATIONS ROUTINE
:*****
001612 112737 000001 002056 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
001620 112737 000001 002054 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
001626 000403
001630 112737 000001 002056 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
001636 $ATYC:
001636 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
001640 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
001642 105737 002054 TSTB $MFLG ;;SHOULD TYPE A MESSAGE?
001646 001450 BEQ 5$ ;;IF NOT: BR
001650 122737 000001 001506 CMPB #APTENV,$ENV ;;OPERATING UNDER APT?
001656 001031 BNE 3$ ;;IF NOT: BR
001660 132737 000100 001507 BITB #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
001666 001425 BEQ 3$ ;;IF NOT: BR
001670 017600 000004 MOV @4(SP),R0 ;;GET MESSAGE ADDR.
001674 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
001702 005737 001466 1$: TST $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
001706 001375 BNE 1$ ;;IF NOT: WAIT
001710 010037 001502 MOV R0,$MSGAD ;;PUT ADDR IN MAILBOX
001714 105720 2$: TSTB (R0)+ ;;FIND END OF MESSAGE
001716 001376 BNE 2$
001720 163700 001502 SUB $MSGAD,R0 ;;SUB START OF MESSAGE
001724 006200 ASR R0 ;;GET MESSAGE LNTH IN WORDS
001726 010037 001504 MOV R0,$MSGGLT ;;PUT LENGTH IN MAILBOX
001732 012737 000004 001466 MOV #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
001740 000413 BR 5$
001742 017637 000004 001766 3$: MOV @4(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
001750 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDRESS
001756 013746 177776 MOV 177776,-(SP) ;;PUSH 177776 ON STACK
001762 004737 044242 JSR PC,$TYPE ;;CALL TYPE MACRO
001766 000000 4$: .WORD 0
001770 5$:
001770 105737 002056 10$: TSTB $FFLG ;;SHOULD REPORT FATAL ERROR?
001774 001416 BEQ 12$ ;;IF NOT: BR
001776 005737 001506 TST $ENV ;;RUNNING UNDER APT?
002002 001413 BEQ 12$ ;;IF NOT: BR
002004 005737 001466 11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
002010 001375 BNE 11$ ;;IF NOT: WAIT
002012 017637 000004 001470 MOV @4(SP),$FATAL ;;GET ERROR #
002020 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
002026 005237 001466 INC $MSGTYPE ;;TELL APT TO TAKE ERROR
002032 105037 002056 12$: CLRB $FFLG ;;CLEAR FATAL FLAG
002036 105037 002055 CLRB $LFLG ;;CLEAR LOG FLAG
002042 105037 002054 CLRB $MFLG ;;CLEAR MESSAGE FLAG
002046 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
002050 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
002052 000207 RTS PC ;;RETURN
002054 000 $MFLG: .BYTE 0 ;;MESSG. FLAG
002055 000 $LFLG: .BYTE 0 ;;LOG FLAG
002056 000 $FFLG: .BYTE 0 ;;FATAL FLAG
.EVEN
000200 APTSIZE-200
000001 APTENV-001
000100 APTSPOOL 100
000040 APTCSUP-040

```

```

581
582
583
584
585 002060 000000
586 002062 000000
587 002064 000000
588 002066 000000
589 002070 000000
590 002072 000000
591
592
593 002074 177570
594 002076 177560
595 002100 177562
596 002102 177564
597 002104 177566
598 002106 000
599 002107 002
600 002110 012
601 002111 000
602 002112 207 377 377
603 002116 077
604 002117 015
605 002120 012 000
606 002122 377 377 000
607
608
609
610
611 177744
612 177746
613 177750
614 177752
615 177754
616 177776
617 000000
618 000001
619 000002
620 000003
621 000004
622 000005
623 000006
624 000007
625 000001
626 000002
627 000004
628 000010
629 000020
630 000040
631 000100
632 000200
633 000400
634 001000
635 002000
636 004000
637 010000

;*****
; USER LABELS
;*****
$STNM: .WORD 0
LOOP: .WORD 0
CMRPAT: .WORD 0
CHRPAT: .WORD 0
FAIL1: .WORD 0
FAIL2: .WORD 0

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

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

```

638	020000	BIT13 = 20000
639	040000	BIT14 = 40000
640	100000	BIT15 = 100000
641		
642		
643	172300	KPDR0 = 172300
644	172302	KPDR1 = 172302
645	172304	KPDR2 = 172304
646	172306	KPDR3 = 172306
647	172310	KPDR4 = 172310
648	172312	KPDR5 = 172312
649	172314	KPDR6 = 172314
650	172316	KPDR7 = 172316
651	172340	KPAR0 = 172340
652	172342	KPAR1 = 172342
653	172344	KPAR2 = 172344
654	172346	KPAR3 = 172346
655	172350	KPAR4 = 172350
656	172352	KPAR5 = 172352
657	172354	KPAR6 = 172354
658	172356	KPAR7 = 172356
659	177572	SR0 = 177572
660	172516	SR3 = 172516
661	170200	UMPR00= 170200
662	170202	UMPR01= 170202
663	170204	UMPR02= 170204
664	170206	UMPR03= 170206
665	170210	UMPR04= 170210
666	170212	UMPR05= 170212
667	170214	UMPR06= 170214
668	170216	UMPR07= 170216
669	170220	UMPR08= 170220
670	170222	UMPR09= 170222
671	170002	BECC = 170002
672	170004	BEBA = 170004
673	170000	BEDA = 170000
674	170006	BECD1 = 170006
675	170016	BECD2 = 170016
676		
677	000200	APTSIZE=200
678	000001	APTENV=001
679	000100	APTSPOOL=100
680	000040	APTCSUP=040
681		
682		:CCR REGISTER
683	000001	DCPI=1
684	000004	FMLO=4
685	000010	FMHI=10
686	000100	WWPD=100
687	000200	PEA=200
688	000400	FC=400
689	001000	UCB=1000
690	002000	WWPT=2000
691	010000	VCIP=10000
692	020000	VSU 20000
693		
694		:CMR REGISTER

695	000001	TDAR=1
696	000002	HODO=2
697	000004	EHA=4
698	000010	AM=10
699	000020	ESA=20
700	000400	HIT=400
701	001000	TPB=1000
702	002000	LPB=2000
703	004000	HPB=4000
704	010000	VLD=10000
705	020000	CM3=20000
706	040000	CM2=40000
707	100000	CM1 100000
708		
709		;CMF REGISTER
710	000040	TPE=40
711	000100	PELO=100
712	000200	PEHI=200
713	100000	CMPE=100000
714		
715	000001	TSTID=1
716	000004	SCPCND=4
717	001015	OFF=1015

718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774

002126 000000
002130 000000
002132 000000
002134 000000
002136 000000
002140 000000
002142 000001

002144 013737 177766 046220
002152 032737 000001 046220
002160 001417
002162 042737 000001 177766
002170 012737 000177 001470
002176 012737 002204 002422
002204 104413
002206 002204
002210 046220 000000
002214 005037 001470
002220 012737 000340 177776
002226 113737 001472 002060
002234 022737 000001 001472
002242 001436
002244 005037 177744
002250 032777 040000 177616
002256 001413
002260 013702 001472
002264 005302
002266 120277 177602
002272 001005
002274 005337 001472
002300 013716 002126
002304 000002
002306 005737 001474
002312 001412
002314 032777 004000 177552
002322 001006
002324 005237 002140
002330 023737 002142 002140
002336 001356
002340 005037 002140
002344 011601
002346 012137 002126
002352 012137 002130
002356 012137 002132
002362 012137 002134

002366 013737 002134 002136
002374 062737 000002 002136

```

*****
          SETUP TEST CONDITIONS:
          1. TEST ITERATIONS
          2. LOAD TEST VECTORS FOR LOOP ON TEST,
             LOOP ON ERROR
*****
STRTST: .WORD 0
STRTLP: .WORD 0
ADRSYNC: .WORD 0
ADRJMP: .WORD 0
ADR1$: .WORD 0
TSTCNT: .WORD 0
TSTIMS: .WORD 1.

$SCPSET: MOV 177766,CPSAVE ;MOVE CPU ERR REG VALUE TO LOC FOR TST ;DPM001
          BIT #BIT00,CPSAVE ;SEE IF THE POWER MONITOR BIT IS ON ;DPM001
          BEQ 2000$ ;BRANCH TO CONTINUE ROUTINE IF CLEAR ;DPM001
          BIC #BIT00,177766 ;CLEAR THE BIT FOUND TO BE SET ;DPM001
          MOV #177,$FATAL ;LET APT KNOW THIS IS A PWR MNTR BIT ERR;DPM001
          MOV #905$,$ERRPC ;MOVE ERROR PC TO $ERRPC ;DPM001
905$: ERROR ;CALL SPECIAL POWER FAIL BIT ERROR CALL ;DPM001
          .WORD -2 ;LOCATION CONTAINING ERROR PC ;DPM001
          .WORD CPSAVE,0 ;LOCATION OF DATA TO PRINT ;DPM001
          CLR $FATAL ;REMOVE 177 FROM $FATAL ;DPM001
2000$: MOV #340,PSW ;CPU HI PRIORITY
          MOVB $TESTN,$STSTNM ;MOVE TEST NUMBER TO $STSTNM
          CMP #1,$TESTN ;IS THIS TEST 1?
          BEQ 3$ ;YES,DO NOT CONSIDER LOOP ON TEST
          CLR CME ;CLEAR THE CACHE MEMORY PARITY FAULT REGISTER
          BIT #BIT14,@SWR ;LOOP ON TEST?
          BEQ 4$ ;NO
          MOV $TESTN,R? ;GET PRESENT TEST NUMBER
          DEC R2 ;GET LAST TEST NUMBER
          CMPB R2,@SWR ;IS THIS THE TEST?
          BNE 4$ ;NO
          DEC $TESTN ;YES;PREPARE FOR LOOP ON TEST
          MOV STRTST,(SP) ;FUDGE RETURN
          RTI ;GO LOOP ON TEST
4$: 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
          CLR TSTCNT
          MOV (SP),R1 ;GET ADDRESS OF FIRST ARGUEMENT
          MOV (R1)+,STRTST ;LOCATION OF START OF TEST
          MOV (R1)+,STRTLP ;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

```

775	002402	012777	000240	177524	MOV	#240,@ADRJMP	:INITIALIZE SLOPE LOCATIONS
776	002410	012777	000240	177520	MOV	#240,@ADR1\$	
777	002416	010116			MOV	RT,(SP)	:SETUP STACK FOR RETURN
778	002420	000002			RTI		
779	002422	000000			\$ERRPC: .WORD	0	:LOCATION TO SAVE ERROR PC

```

780
781
782
783
784
785
786 002424 012701 060000
787 002430 012402
788 002432 012421
789 002434 020402
790 002436 001375
791 002440 013721 002450
792 002444 000137 060000
793 002450 000204
794 002452 012701 070000
795 002456 012402
796 002460 012421
797 002462 020402
798 002464 001375
799 002466 013721 002476
800 002472 000137 070000
801 002476 000204
    
```

```

.SBTTL RELOCATION HANDLERS
:*****
: RELOCATION HANDLERS
:*****
RELCTL: MOV #LOW1,R1 ;START OF LOW SPACE
        MOV (R4)+,R2 ;END OF MOVE
1$: MOV (R4)+,(R1)+ ;TRANSFER TEST
    CMP R4,R2 ;PROCEED TO STOP MARK
    BNE 1$
    MOV 2$(R1)+ ;RETURN INSTRUCTION
    JMP 60000 ;START TESTS
2$: RTS R4
RELCTH: MOV #HIGH1,R1 ;START OF HI SPACE
        MOV (R4)+,R2 ;END OF MOVE
1$: MOV (R4)+,(R1)+ ;TRANSFER TEST
    CMP R4,R2 ;PROCEED TO STOP MARK
    BNE 1$
    MOV 2$(R1)+ ;RETURN INSTRUCTION
    JMP 70000 ;START TESTS
2$: RTS R4
    
```

```
802 002500 012706 000500      BEGIN:  MOV    #500,SP      ;SET UP STACK
803 002504 000005              RESET
804 002506 012737 000340 177776  MOV    #340,PSW      ;CPU HI PRIORITY
805 002514 012737 001015 177746  MOV    #OFF,CCR      ;DISABLE CACHE
806 002522 005037 001472              CLR    $TESTN        ;RESET TEST ID COUNTER
807 002526 012737 000002 000000  MOV    #2,0          ;INITIALIZE A FEW VECTORS
808 002534 005037 000002              CLR    2
809 002540 012737 000006 000004  MOV    #6,4
810 002546 005037 000006              CLR    6
811 002552 012737 000116 000114  MOV    #16,114
812 002560 005037 000116              CLR    116
813 002564 005037 002062              CLR    LOOP          ;SOFTWARE DELAY
814 002570 005337 002062      1$:   DEC    LOOP
815 002574 001375              BNE    1$
```


820

```

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

```

```

002576 000004
002600 002610
002602 002610
002604 000000
002606 002634
002610
821 002610 012737 002642 000004
822 002616 012737 000340 000006
823 002624 005737 177744
824 002630 000240
825 002632 000240
826 002634 000240
002636 000240
827 002640 000411
828 002642 022626
829 002644 012737 000006 000004
830 002652 005037 000006
831 002656 104413

002660 002656
832
833
834
835
836 002662 000000
837 002664 000240
002666 005237 001472

TST1:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION

40$:
1$: MOV #2$,4 ;SETUP TRAP VECTOR
MOV #340,6
TST CME ;READ PARITY REGISTER
NOP
NOP
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BR 10$ ;NO FAULT;GO TO NEXT TEST
2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
MOV #6,4 ;RESTORE TRAP VECTOR
CLR 6
ERROR ;ERROR
;-----
;CACHE REGISTER RESPONSE TESTS
;READING PARITY FAULT REGISTER
;CAUSED TIMEOUT

;WORD .-2

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

```

842

```

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

```

```

002672
002672 000004
002674 002704
002676 002704
002700 000000
002702 002730
002704
843 002704 012737 002736 000004
844 002712 012737 000340 000006
845 002720 005737 177746
846 002724 000240
847 002726 000240
848 002730 000240
002732 000240
849 002734 000411
850 002736 022626
851 002740 012737 000006 000004
852 002746 005037 000006
853 002752 104413
002754 002752
854
855
856
857
858 002756 000000
859 002760 000240
002762 005237 001472

```

```

TST2:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
.WORD 40$ ;LOOP ON ERROR START LOCATION
.WORD 1$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 0 ;LOOP ON ERROR END LOCATION
.WORD 25$
40$:
1$: MOV #2$,4 ;SETUP TRAP VECTOR
MOV #340,6
TST CCR ;READ CACHE CONTROL REGISTER
NOP
NOP
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BR 10$ ;NO FAULT;GO TO NEXT TEST
2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
MOV #6,4 ;RESTORE TRAP VECTOR
CLR 6
ERROR ;ERROR
;-----
.WORD -2
;CACHE REGISTER RESPONSE TESTS
;READING CACHE CONTROL REGISTER
;CAUSED TIMEOUT
.WORD 0
10$: NOP ;END OF TEST
INC $TESTN ;INCREMENT TEST COUNTER

```

TEST # 3 - READ CMR TO CHECK ADDRESS SELECT LOGIC

004

```

.SBTTL TEST # 3 - READ CMR TO CHECK ADDRESS SELECT LOGIC
:*****
:*TEST 3      READ CMR TO CHECK ADDRESS SELECT LOGIC
:*          ATTEMPT READ INTO CMR TO CHECK ADDRESS SELECT LOGIC
:*          IF TIME OCCURES THEN LOGIC IN FAULT
:*****
TST3:
    
```

```

002766
002766 000004
    SPCOND
    :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
    :ERROR/LOOP ON TEST
    :TEST START LOCATION
002770 003000
002772 003000
002774 000000
002776 003024
    :LOOP ON ERROR START LOCATION
    :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
    :LOOP ON ERROR END LOCATION
003000
865 003000 012737 003032 000004 40$:
866 003006 012737 000340 000006 1$:
867 003014 005737 177750
868 003020 000240
869 003022 000240
870 003024 000240
871 003030 000411
872 003032 022626
873 003034 012737 000006 000004
874 003042 005037 000006
875 003046 104413
    MOV #2$,4
    MOV #340,6
    TST CMR
    NOP
    NOP
    :SETUP TRAP VECTOR
    :READ MAINTENANCE REGISTER
    :INSTRUCTION 'JMP 1$' PLACED HERE
    :FOR LOOP ON ERROR
    :NO FAULT;GO TO NEXT TEST
    :READJUST STACK DUE TO INTERRUPT
    :RESTORE TRAP VECTOR
    BR 10$
    CMP (SP)+,(SP)+
    MOV #6,4
    CLR 6
    :ERROR
    :-----
    :WORD .-2
    :CACHE REGISTER RESPONSE TESTS
    :READING MAINTENANCE REGISTER
    :CAUSED TIMEOUT
003050 003046
876
877
878
879 003052 000000
880 003054 000240
003056 005237 001472
    :END OF TEST
    :INCREMENT TEST COUNTER
    :WORD 0
    :NOP
    :INC $TESTN
    
```

885

```

.SBTTL TEST # 4 - READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
:*****
:TEST 4      READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
:*          ATTEMPT READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
:*          IF TIME OUT OCCURES THEN LOGIC IN FAULT
:*****
TST4:
      SCPCND          :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                        :ERROR/LOOP ON TEST
      .WORD 40$       :TEST START LOCATION
      .WORD 1$        :LOOP ON ERROR START LOCATION
      .WORD 0         :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$      :LOOP ON ERROR END LOCATION
40$:
886 003074 012737 003126 000004 1$: MOV #2$,4          :SETUP TRAP VECTOR
887 003102 012737 000340 000006 MOV #340,6
888 003110 005737 177752 TST CHR          :READ HIT REGISTER
889 003114 000240 NOP
890 003116 000240 NOP
891 003120 000240 25$: NOP          :INSTRUCTION 'JMP 1$' PLACED HERE
      .WORD 000240 :FOR LOOP ON ERROR
892 003124 000411 BR 10$          :NO FAULT;GO TO NEXT TEST
893 003126 022626 2$: CMP (SP)+,(SP)+ :READJUST STACK DUE TO INTERRUPT
894 003130 012737 000006 000004 MOV #6,4          :RESTORE TRAP VECTOR
895 003136 005037 000006 CLR 6
896 003142 104413 ERROR          :ERROR
      .WORD -2      :-----
897 003144 003142 .WORD -2          :CACHE REGISTER RESPONSE TESTS
898 :READING HIT REGISTER
899 :CAUSED TIMEOUT
900 003146 000000 .WORD 0
901 003150 000240 10$: NOP          :END OF TEST
      .WORD 000240 :INCRMENT TEST COUNTER
      .WORD 005237 001472 INC $TESTN
    
```

906

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

```
TST5:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMEN ED FOR THIS TEST
;LOOP ON ERROR END LOCATION

40$:
1$: MOV #2$,4 ;SETUP TRAP VECTOR
MOV #340,6
TST CDR ;READ DATA REGISTER
NOP
NOP
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BR 10$ ;NO FAULT;GO TO NEXT TEST
2$: CMP (SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
MOV #6,4 ;RESTORE TRAP VECTOR
CLR 6
ERROR ;ERROR
;-----

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

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

```
003156
003156 000004
003160 003170
003162 003170
003164 000000
003166 003214
003170
907 003170 012737 003222 000004
908 003176 012737 000340 000006
909 003204 005737 177754
910 003210 000240
911 003212 000240
912 003214 000240
003216 000240
913 003220 000411
914 003222 022626
915 003224 012737 000006 000004
916 003232 005037 000006
917 003236 104413
003240 003236
918
919
920
92 003242 000000
922 003244 000240
003246 005237 001472
```

930

```

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

```

```

003252
003252 000004
:
003254 003264
003256 003264
003260 000000
003262 003272
003264
931 003264 012737 000001 177744
932 003272 000240
003274 000240
933 003276 032737 000001 177744
934 003304 001403
935 003306 104413
:
003310 003306
936
937
938
939 003312 000000
940 003314 000240
003316 005237 001472
:
TST6:
SCPCND
:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
:
40$:
1$: MOV #1,CME :WRITE 1 INTO BIT00
25$: NOP :INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:CHECK FOR 0
:PASS;NEXT TEST
:ERROR
:-----
:
:WORD .-2
:
:
:
:WORD 0
10$: NOP :END OF TEST
INC $TESTN :INCREMENT TEST COUNTER

```

4.7

```

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

```

```

003322 000004
003324 003334
003326 003334
003330 000000
003332 003342
003334
948 003334 032737 000001 177746 40$:
949 003342 000240 1$:
003344 000240 25$:
950 003346 001003
951 003350 104413
003352 003350
952
953
954 003354 000000
955 003356 000240 10$:
003360 005237 001472
          SCPCND
          .WORD 40$
          .WORD 1$
          .WORD 0
          .WORD 25$
          BIT #1,CCR
          NOP
          BNE 10$
          ERROR
          .WORD -2
          .WORD 0
          NOP
          INC $TESTN
          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          ;TEST START LOCATION
          ;LOOP ON ERROR START LOCATION
          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          ;LOOP ON ERROR END LOCATION
          ;CHECK BIT00 FOR 1
          ;INSTRUCTION 'JMP 1$' PLACED HERE
          ;FOR LOOP ON ERROR
          ;PASS:NXT TEST
          ;ERROR
          ;-----
          ;CACHE REGISTER RESPONSE TESTS
          ;WROTE 1 INTO BIT00 CCR; READ 0
          ;END OF TEST
          ;INCREMENT TEST COUNTER

```

960

..SBTTL TEST # 10 - CACHE CONTROL REGISTER DATA TEST (CCR)

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

TST10:

003364	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
003366	003376			.WORD	40\$:ERROR/LOOP ON TEST
003370	003376			.WORD	1\$:TEST START LOCATION
003372	000000			.WORD	0	:LOOP ON ERROR START LOCATION
003374	003404			.WORD	25\$:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
003376						:LOOP ON ERROR END LOCATION
961	003376	032737	010000 177746	40\$:		
962	003404	000240		1\$:	BIT #BIT12.CCR	:CHECK FOR 0
	003406	000240		25\$:	NOP	:INSTRUCTION 'JMP 1\$' PLACED HERE
963	003410	001403			NOP	:FOR LOOP ON ERROR
964	003412	104413			BEQ 10\$:PASS
					ERROR	:ERROR
						:-----
	003414	003412			.WORD -2	
965						:CCR DATA TEST
966						:READ 1 FROM CCR BIT12. A CLEARING OF
967						:VALID STORE AT THIS TIME SHOULD NOT
968						:BE INDICATED
969	003416	000000			.WORD 0	
970	003420	000240		10\$:	NOP	:END OF TEST
	003422	005237	001472		INC \$TESTN	:INCREMENT TEST COUNTER

976

```

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

```

```

003426
003426 000004
003430 003440
003432 003446
003434 000000
003436 003460
003440
977 003440 012737 001415 177746 40$: MOV #OFF+BIT08,CCR ;DISABLE AND FLUSH CACHE
978 003446 042737 000001 177746 1$: BIC #BIT00,CCR ;WRITE 0
979 003454 013700 177746 MOV CCR,R0 ;SAVE CCR CONTENTS
980 003460 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
003462 000240 NOP ;FOR LOOP ON ERROR
981 003464 032700 000001 BIT #BIT00,R0 ;CHECK FOR 0
982 003470 001403 BEQ 10$ ;PASS; NXT TEST
983 003472 104413 ERROR ;ERROR
;-----
003474 003472 .WORD .-2 ;CCR DATA TEST
984 ;WROTE 0 INTO BIT00 CCR; READ 1
985
986 003476 000000 .WORD 0
987 003500 000240 10$: NOP ;END OF TEST
003502 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```

993

```
.SBTTL TEST # 12 - TEST CLEARING OF BIT 2 OF CCR
:*****
:*TEST 12 TEST CLEARING OF BIT 2 OF CCR
:* WRITE ZERO INTO CCR BIT02(FMLO) THEN READ CCR
:* IF BIT02 IS READ AS ONE THEN CCR REGISTER MAY BE BAD
:* OR CACHE REGISTER DATA PATH MAY BE A1 FAULT
:*****
```

```
003506
003506 000004          SCPCND          ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
003510 003520          .WORD 40$          ;TEST START LOCATION
003512 070000          .WORD 1$-40$+67764        ;LOOP ON ERROR START LOCATION
003514 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
003516 070006          .WORD 25$-40$+67764    ;LOOP ON ERROR END LOCATION
003520 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
003526 004437 002452      JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
003532 003576          .WORD 10$+2        ;ADDRESS OF START OF NEXT TEST
```

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

```
994 003534 042737 000004 177746 1$: BIC #BIT02,CCR ;WRITE 0
995 003542 000240          25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      003544 000240          NOP ;FOR LOOP ON ERROR
996 003546 013701 177746      MOV CCR,R1 ;SAVE CCR CONTENTS
997 003552 012737 001015 177746      MOV #OFF,CCR ;DISABLE CACHE
998 003560 032701 000004      BIT #BIT02,R1 ;CHECK FOR 0
999 003564 001403          BEQ 10$ ;PASS; NXT TEST
1000 003566 104413          ERROR ;ERROR
                                ;-----
      003570 003566          .WORD -.2 ;CCR DATA TEST
1001
1002
1003 003572 000000          .WORD 0 ;WROTE 0 INTO CCR BIT02; READ 1
1004 003574 000240          10$: NOP ;END OF TEST
      003576 005237 001472      INC $TESTN ;INCREMENT TEST COUNTER
```

1010

```

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

```

```

003602 000004
003604 003614
003606 003614
003610 000000
003612 003622
003614
1011 003614 052737 000004 177746 40$:
1012 003622 000240 1$:
003624 000240 25$:
1013 003626 032737 000004 177746
1014 003634 001003
1015 003636 104413
003640 003636
1016
1017
1018 003642 000000
1019 003644 000240 001472
003646 005237

SCPCND
:SCOPE CONDITIONS:GO SFT 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

BIS #FMLO,CCR
:INSTRUCTION 'JMP 1$' PLACED HERE
NOP
:FOR LOOP ON ERROR
NOP
BIT #BIT02,CCR
:CHECK FOR 1
BNE 10$
:PASS
ERROR
:ERROR
:-----

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

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

```

1025

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

003652
 003652 000004

003654 003664
 003656 060000
 003660 000000
 003662 060012
 003664 012737
 003672 004437
 003676 003742

001015 177746
 002424

```
TST14: SCPCND ;SCOPE CONDITIONS:GO SFT 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
```

1026 003700 042737 000010 177746 1\$:
 1027 003706 013700 177746
 1028 003712 000240 25\$:
 003714 000240
 1029 003716 012737 001015 177746
 1030 003724 032700 000010
 1031 003730 001403
 1032 003732 104413
 003734 003732
 1033
 1034
 1035 003736 000000
 1036 003740 000240 10\$:
 003742 005237 001472

```
BIC #BIT03,CCR ;WRITE 0
MOV CCR,R0 ;SAVE CONTENTS OF CCR
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
MOV #OFF,CCR ;;DISABLE CACHE
BIT #BIT03,R0 ;CHECK FOR 0
BEG 10$ ;PASS
ERROR ;ERROR
;-----
;CCR DATA TEST
;WROTE 0 INTO CCR BIT03; READ 1
;END OF TEST
;INCREMENT TEST COUNTER
```

1042

```
.SBTTL TEST # 15 - TEST SETTING OF CCR BIT 3
*****
:TEST 15 TEST SETTING OF CCR BIT 3
:* WRITE 1 INTO CCR BIT03(FMHI) AND ASSURE IT READS 1.
:* IF CCR BIT03 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
:* OR CACHE REGISTER DATA PATH MAY BE AT FAULT
*****
TST15:
```

```
003746 000004
003750 003760
003752 003760
003754 000000
003756 003766
003760
1043 003760 052737 000010 177746 40$:
1044 003766 000240 1$:
003770 000240 25$:
1045 003772 032737 000010 177746
1046 004000 001003
1047 004007 104413
004004 004002
1048
1049
1050 004006 000000
1051 004010 000240 10$:
004012 005237 001472

SCPCND ;SCOPE CONDITIONS:GO SFT 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

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

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

;END OF TEST
;INCREMENT TEST COUNTER

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

BIS #FMHI,CCR
NOP
NOP
BIT #BIT03,CCR
BNE 10$
ERROR

.WORD -2

.WORD 0
NOP
INC $TESTN
```

1057

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

```
TST16:
          004016 000004          SCPCWD          ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
          004016 000004          ;ERROR/LOOP ON TEST
          004020 004030          .WORD 40$          ;TEST START LOCATION
          004022 004030          .WORD 1$          ;LOOP ON ERROR START LOCATION
          004024 000000          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          004026 004036          .WORD 25$         ;LOOP ON ERROR END LOCATION
          004030 004030          40$:          ;WRITE 0
1058 004030 042737 000100 177746 1$: BIC #BIT06,CCR ;INSTRUCTION 'JMP 1$' PLACED HERE
1059 004036 000240          25$: NOP          ;FOR LOOP ON ERROR
          004040 000240          NOP          ;CHECK FOR 0
1060 004042 032737 000100 177746 BIT #BIT06,CCR ;PASS
1061 004050 001403          BEQ 10$         ;ERROR
1062 004052 104413          ERROR          ;-----
          004054 004052          .WORD -2         ;CCR DATA TEST
1063 004054 004052          ;WROTE 0 INTO CCR BIT06; READ 1
1064 004054 004052          ;CCR DATA TEST
          004056 000000          .WORD 0           ;WROTE 0 INTO CCR BIT06; READ 1
1065 004060 000240          10$: NOP          ;END OF TEST
1066 004062 005237 001472          INC $TESTN      ;INCREMENT TEST COUNTER
```

1072

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

```
TST17:
004066 000004          SCPLND          ;SCOPE CONDITIONS.GO SFT UP FOR LOOP ON
004066 000004          .WORD 40$          ;ERROR/LOOP ON TEST
004070 004100          .WORD 1$          ;TEST START LOCATION
004072 004100          .WORD 0           ;LOOP ON ERROR START LOCATION
004074 000000          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004076 004106          .WORD 25$         ;LOOP ON ERROR END LOCATION
004100 40$:
1073 004100 042737 000200 177746 1$: BIC #BIT07,CCR ;WRITE 0
1074 004106 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
004110 000240 NOP ;FOR LOOP ON ERROR
1075 004112 032737 000200 177746 BIT #BIT07,CCR ;CHECK FOR 0
1076 004120 001403 BEQ 10$ ;PASS
1077 004122 104413 ERROR ;ERROR
;-----
004124 004122 .WORD -2 ;CCR DATA TEST
1078 ;WROTE 0 INTO CCR BIT07; READ 1
1079 ;END OF TEST
1080 004126 000000 .WORD 0 ;INCREMENT TEST COUNTER
1081 004130 000240 10$: NOP
004132 005237 001472 INC $TESTN
```

087

```

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

```

```

004136
004136 000004
004140 004150
004142 004150
004144 000000
004146 004162
004150
1088 004150 052737 000200 177746 40$:
1089 004156 013700 177746 1$: BIS #BIT07,CCR ;WRITE 1
1090 004162 000240 25$: MOV CCR,RO ;SAVE CCR CONTENTS
004164 000240 ;INSTRUCTION 'JMP 1$' PLACED HERE
1091 004166 012737 001015 177746 NOP ;FOR LOOP ON ERROR
1092 004174 032700 000200 MOV #OFF,CCR ;DISABLE CACHE
1093 004200 001003 BIT #BIT07,RO ;CHECK FOR 1
1094 004202 104413 BNE 10$ ;PASS
;ERROR
;-----

004204 004202 .WORD -2 ;CCR DATA TEST
1095 ;WROTE 1 INTO CCR BIT07; READ 0
1096
1097 004206 000000 .WORD 0
1098 004210 000240 10$: NOP ;END OF TEST
004212 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```


1104

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

```
TST21:
004216
004216 000004
004220 004230
004222 004230
004224 000000
004226 004236
004230
1105 004230 042737 000400 177746 40$:
1106 004236 000240 1$:
004240 000240 25$:
1107 004242 032737 000400 177746
1108 004250 001401
1109
1110
1111 004252 000000
1112 004254 000240 10$:
004256 005237 001472
SCPCND
:SCOPE CONDITIONS:GO SFT 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
BIC #BIT08,CCR :WRITE 0
NOP :INSTRUCTION 'JMP 1$' PLACED HERE
NOP :FOR LOOP ON ERROR
BIT #BIT08,CCR :CHECK FOR 0
BEQ 10$ :PASS
:CCR DATA TEST
:WROTE 0 INTO CCR BIT08; READ 1
:WORD 0
NOP :END OF TEST
INC $TESTN :INCREMENT TEST COUNTER
```

1118

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

```
TST22:
        SCPCND                ;SCOPE CONDITIONS:GO SFT JP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
        .WORD 40$              ;LOOP ON ERROR START LOCATION
        .WORD 1$              ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 0                ;LOOP ON ERROR END LOCATION
        .WORD 25$
1119 004274 052737 001000 177746 40$: BIS #UCB,CCR ;
1120 004302 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
        004304 000240 NOP ;FOR LOOP ON ERROR
1121 004306 032737 001000 177746 BIT #BIT09,CCR ;CHECK FOR 1
1122 004314 001003 BNE 10$ ;PASS
1123 004316 104413 ERROR ;ERROR
                                ;-----
        .WORD .-2                ;CCR DATA TEST
                                ;WROTE 1 INTO CCR BIT09; READ 0
1124 004320 004316
1125
1126 004322 000000 .WORD 0
1127 004324 000240 10$: NOP ;END OF TEST
        004326 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
```

1131

```

.SBTTL TEST # 23 - TEST CLEARING BIT 10 OF CCR
:*****
:*TEST 23 TEST CLEARING BIT 10 OF CCR
:* WRITE ZERO INTO CCR BIT10(WWPT) AND READ 0
:*****
TST23:

```

004332	000004			SCPCND		:SCOPE CONDITIONS GO SET UP FOR LOOP ON
004334	004344			.WORD 40\$:ERROR/LOOP ON TEST
004336	004344			.WORD 1\$:TEST START LOCATION
004340	000000			.WORD 0		:LOOP ON ERROR START LOCATION
004342	004352			.WORD 25\$:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004344						:LOOP ON ERROR END LOCATION
1132	004344	042737	002000 177746	40\$: BIC #BIT10,CCR		:WRITE 0
1133	004352	000240		1\$: NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	004354	000240		25\$: NOP		:FOR LOOP ON ERROR
1134	004356	032737	002000 177746	BIT #BIT10,CCR		:CHECK FOR 0
1135	004364	001403		BEQ 10\$:PASS
1136	004366	104413		ERROR		:ERROR
						:-----
	004370	004366		.WORD -2		:CCR DATA TEST
1137						:WROTE 0 INTO CCR BIT 10; READ 1
1138						
1139	004372	000000		.WORD 0		
1140	004374	000240		10\$: NOP		:END OF TEST
	004376	005237	001472	INC \$TESTN		:INCREMENT TEST COUNTER

1145

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

```
TST24:
004402          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
004402 000004          ;ERROR/LOOP ON TEST
004404 004414          ;TEST START LOCATION
004406 004414          ;LOOP ON ERROR START LOCATION
004410 000000          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
004412 004422          ;LOOP ON ERROR END LOCATION
004414          40$:
1146 004414 052737 000002 177746 1$: BIS #BIT01,CCR ;WRITE 1 INTO UNUSED BIT
1147 004422 000240          25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
004424 000240          ;FOR LOOP ON ERROR
1148 004426 032737 000002 177746 BIT #BIT01,CCR ;CHECK THAT BIT READS 0
1149 004434 001403          ;PASS
1150 004436 104413          ;ERROR
004440 004436          ;-----
1151          .WORD -2          ;CCR UNUSED BIT TEST
1152          ;READ 1 FROM UNUSED CCR BIT01
1153          ;SHOULD READ 0
1154 004442 000000          .WORD 0
1155 004444 000240          10$: NOP ;END OF TEST
004446 005237 001472          INC $TESTN ;INCREMENT TEST COUNTER
```

```

004452 000004
004454 004464
004456 004464
004460 000000
004462 004472
004464
1161 004464 052737 000020 177746
1162 004472 000240
004474 000240
1163 004476 032737 000020 177746
1164 004504 001403
1165 004506 104413
004510 004506
1166
1167
1168
1169 004512 000000
1170 004514 000240
004516 005237 001472

```

```

.SBTTL TEST # 25 - TEST UNUSED BIT 4 OF CCR
*****
*TEST 25 TEST UNUSED BIT 4 OF CCR
* WRITE ONE INTO UNUSED CCR BIT04 THEN READ CCR
* IF CCR BIT04 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
*****
TST25:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
      .WORD 40$            ;TEST START LOCATION
      .WORD 1$            ;LOOP ON ERROR START LOCATION
      .WORD 0              ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$           ;LOOP ON ERROR END LOCATION
40$:  BIS #BIT04,CCR        ;WRITE 1 INTO UNUSED BIT
1$:   NOP                  ;INSTRUCTION 'JMP 1$' PLACED HERE
25$:  NOP                  ;FOR LOOP ON ERROR
      BIT #BIT04,CCR        ;CHECK THAT BIT READS 0
      BEQ 10$              ;PASS
      ERROR                 ;ERROR
      ;-----
      .WORD -2             ;CCR UNUSED BIT TEST
                          ;READ 1 FROM UNUSED CCR BIT04
                          ;SHOULD READ 0
10$:  .WORD 0              ;END OF TEST
      NOP                  ;INCREMENT TEST COUNTER
      INC $TESTN

```

1175

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

```
TST26:
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
004522 000004
004524 004534 .WORD 40$
004526 004534 .WORD 1$
004530 000000 .WORD 0
004532 004542 .WORD 25$
004534 40$:
1176 004534 052737 000040 177746 1$: BIS #BIT05,CCR ;WRITE 1 INTO UNUSED BIT
1177 004542 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
004544 000240 NOP ;FOR LOOP ON ERROR
1178 004546 032737 000040 177746 BIT #BIT05,CCR ;CHECK THAT BIT READS 0
1179 004554 001403 BEQ 10$ ;PASS
1180 004556 104413 ERROR ;ERROR
;-----
004560 004556 .WORD -2
1181 ;CCR UNUSED BIT TEST
1182 ;READ 1 FROM UNUSED CCR BIT05
1183 ;SHOULD READ 0
1184 004562 000000 .WORD 0
1185 004564 000240 10$: NOP ;END OF TEST
004566 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
```

1190

```

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

```

```

004572
004572 000004
004574 004604
004576 004604
004600 000000
004602 004612
004604
1191 004604 052737 000400 177746 40$:
1192 004612 000240 1$:
004614 000240 25$:
1193 004616 032737 000400 177746
1194 004624 001403
1195 004626 104413
004630 004626
1196
1197
1198
1199 004632 000000
1200 004634 000240 10$:
004636 005237 001472

```

```

TST27:
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
;WRITE 1 INTO UNUSED BIT
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;CHECK THAT BIT READS 0
;PASS
;ERROR
;-----
;CCR UNUSED BIT TEST
;READ 1 FROM UNUSED CCR BIT08
;SHOULD READ 0
;END OF TEST
;INCREMENT TEST COUNTER

```

1206

```

.SBTTL TEST # 30 - TEST UNUSED BIT 11 OF CCR
*****
*TEST 30 TEST UNUSED BIT 11 OF CCR
* WRITE ONE INTO UNUSED CCR BIT11 THEN READ CCR
* IF CCR BIT11 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
*****

```

```

004642 000004
004644 004654
004646 004654
004650 000000
004652 004662
004654
1206 004654 052737 004000 177746 40$: BIS #BIT11,CCR ;WRITE 1 INTO UNUSED BIT
1207 004662 000240 1$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
004664 000240 25$: NOP ;FOR LOOP ON ERROR
1208 004666 032737 004000 177746 BIT #BIT11,CCR ;CHECK THAT BIT READS 0
1209 004674 001403 BEQ 10$ ;PASS
1210 004676 104413 ERROR ;ERROR
;-----
004700 004676 .WORD -2 ;CCR UNUSED BIT TEST
1211 ;READ 1 FROM UNUSED CCR BIT11
1212 ;SHOULD READ 0
1213
1214 004702 000000 .WORD 0
1215 004704 000240 10$: NOP ;END OF TEST
004706 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```


1220

```

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

```

```

004712
004712 000004
004714 004724
004716 004724
004720 000000
004722 004732
004724
1221 004724 052737 040000 177746 40$:
1222 004732 000240 1$: BIS #BIT14,CCR ;WRITE 1 INTO UNUSED BIT
004734 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
1223 004736 032737 040000 177746 BIT #BIT14,CCR ;FOR LOOP ON ERROR
1224 004744 001403 BEQ 10$ ;CHECK THAT BIT READS 0
1225 004746 104413 ERROR ;PASS
;ERROR
;-----
004750 004746 .WORD -2 ;CCR UNUSED BIT TEST
1226 ;READ 1 FROM UNUSED CCR BIT14
1227 ;SHOULD READ 0
1228
1229 004752 000000 .WORD 0
1230 004754 000240 10$: NOP ;END OF TEST
004756 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```

1235

```

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

```

004762
004762 000004
004764 004774
004766 004774
004770 000000
004772 005002
004774
1236 004774 052737 100000 177746 40$:
1237 005002 000240 1$: BIS #BIT15,CCR ;WRITE 1 INTO UNUSED BIT
005004 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
1238 005006 032737 100000 177746 NOP ;FOR LOOP ON ERROR
1239 005014 001403 BIT #BIT15,CCR ;CHECK THAT BIT READS 0
1240 005016 104413 BEQ 10$ ;PASS
ERROR ;ERROR
;-----
005020 005016 .WORD -2 ;CCR UNUSED BIT TEST
1241 ;READ 1 FROM UNUSED CCR BIT15
1242 ;SHOULD READ 0
1243
1244 005022 000000 .WORD 0
1245 005024 000240 10$: NOP ;END OF TEST
005026 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
    
```

1250

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

```
005032
005032 000004
005034 005044
005036 005044
005040 000000
005042 005052
005044
1251 005044 052737 000001 177744
1252 005052 000240
005054 000240
1253 005056 032737 000001 177744
1254 005064 001403
1255 005066 104413
005070 005066
1256
1257
1258 005072 000000
1259 005074 000240
005076 005237 001472
```

```
TST33:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
.WORD 40$ ;LOOP ON ERROR START LOCATION
.WORD 1$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 0 ;LOOP ON ERROR END LOCATION
.WORD 25$
40$: BIS #BIT00,CME ;WRITE 1 INTO BIT00
1$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
25$: NOP ;FOR LOOP ON ERROR
BIT #BIT00,CME ;CHECK FOR 0
BEQ 10$ ;PASS
ERROR ;ERROR
;-----
.WORD .-2 ;CME UNUSED BIT TEST
;READ 1 FROM UNUSED CME BIT00
.WORD 0
10$: NOP ;END OF TEST
INC $TESTN ;INCREMENT TEST COUNTER
```

1260

.SBTTL TEST # 34 - CME UNUSED BIT 1 TEST
 :*****
 :*TEST 34 CME UNUSED BIT 1 TEST
 :*****
 TST34:

	005102	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
	005104	005114			.WORD 40\$:ERROR/LOOP ON TEST
	005106	005114			.WORD 1\$:TEST START LOCATION
	005110	000000			.WORD 0		:LOOP ON ERROR START LOCATION
	005112	005122			.WORD 25\$:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
	005114			40\$:			:LOOP ON ERROR END LOCATION
1261	005114	052737	000002	177744	1\$:	BIS #BIT01,CME	:WRITE 1 INTO BIT01
1262	005122	000240			25\$:	NOP	:INSTRUCTION 'JMP 1\$' PLACED HERE
	005124	000240				NOP	:FOR LOOP ON ERROR
1263	005126	032737	000002	177744		BIT #BIT01,CME	:CHECK FOR 0
1264	005134	001403				BEQ 10\$:PASS
1265	005136	104413				ERROR	:ERROR
	005140	005136				.WORD -2	:-----
1266							:CME UNUSED BIT TEST
1267							:READ 1 FROM UNUSED CME BIT01
1268	005142	000000				.WORD 0	
1269	005144	000240			10\$:	NOP	:END OF TEST
	005146	005237	001472			INC \$TESTN	:INCREMENT TEST COUNTER

..SHTTL TEST # 35 - CME UNUSED BIT 2 TEST

.....
 :TEST 35 CME UNUSED BIT 2 TEST

TEST35:

005152	005152	000004		SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
						:ERROR/LOOP ON TEST
005154	005164			.WORD	40\$:TEST START LOCATION
005156	005164			.WORD	1\$:LOOP ON ERROR START LOCATION
005160	000000			.WORD	0	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
005162	005172			.WORD	25\$:LOOP ON ERROR END LOCATION
005164						
1271	005164	052737	000004 177744	40\$:		:WRITE 1 INTO BIT02
1272	005172	000240		1\$:	BIS #BIT02,CME	:INSTRUCTION 'JMP 1\$' PLACED HERE
	005174	000240		25\$:	NOP	:FOR LOOP ON ERROR
1273	005176	032737	000004 177744		NOP	:CHECK FOR 0
1274	005204	001403			BIT #BIT02,CME	:PASS
1275	005206	104413			BEQ 10\$:ERROR
					ERROR	:-----
	005210	005206			.WORD	.-2
1276						:CME UNUSED BIT TEST
1277						:READ 1 FROM UNUSED CME BIT02
1278	005212	000000			.WORD	0
1279	005214	000240		10\$:	NOP	:END OF TEST
	005216	005237	001472		INC \$TESTN	:INCREMENT TEST COUNTER

1290

```

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

```

```

005272      000004
005274      005304
005276      005304
005300      000000
005302      005312
005304
1291 005304 052737 000020 177744 40$:
1292 005312 000240          1$:
005314      000240          25$:
1293 005316 032737 000020 177744
1294 005324 001403
1295 005326 104413
C05330      005326
1296
1297
1298 005332 000000
1299 005334 000240          10$:
005336      005237 001472

```

```

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
BIS #BIT04,CME :WRITE 1 INTO BIT04
NOP :INSTRUCTION 'JMP 1$' PLACED HERE
NOP :FOR LOOP ON ERROR
BIT #BIT04,CME :CHECK FOR 0
BEQ 10$ :PASS
ERROR :ERROR
:-----
.WORD .-2
: CME UNUSED BIT TEST
:READ 1 FROM UNUSED CME BIT04
.WORD 0
NOP
INC $TESTN :END OF TEST
:INCREMENT TEST COUNTER

```

1300

```

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

```

005342	005342	000004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON	
							:ERROR/LOOP ON TEST	
005344	005344	005354			.WORD	40\$:TEST START LOCATION	
005346	005346	005354			.WORD	1\$:LOOP ON ERROR START LOCATION	
005350	005350	000000			.WORD	0	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST	
005352	005352	005362			.WORD	25\$:LOOP ON ERROR END LOCATION	
				40\$:				
1301	005354	052737	000400	177744	1\$:	BIS	#BIT08,CME	:WRITE 1 INTO BIT08
1302	005362	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	005364	000240				NOP		:FOR LOOP ON ERROR
1303	005366	032737	000400	177744		BIT	#BIT08,CME	:CHECK FOR 0
1304	005374	001403				BEQ	10\$:PASS
1305	005376	104413				ERROR		:ERROR
								:-----
	005400	005376			.WORD	.-2		:CME UNUSED BIT TEST
1306								:READ 1 FROM UNUSED CME BIT08
1307								
1308	005402	000000			.WORD	0		
1309	005404	000240			10\$:	NOP		:END OF TEST
	005406	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER


```

.SBTTL TEST # 41 - CME UNUSED BIT 9 TEST
:*****
:TEST 41 CME UNUSED BIT 9 TEST
:*****
*ST41:
    
```

```

005412
005412 000004
005414 005424
005416 005424
005420 000000
005422 005432
005424
1311 005424 052737 001000 177744 40$:
1312 005432 000240 1$:
005434 000240 25$:
1313 005436 032737 001000 177744
1314 005444 001403
1315 005446 104413
005450 005446
1316
1317
1318 005452 000000
1319 005454 000240 10$:
005456 005237 001472
    
```

```

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
        ;WRITE 1 INTO BIT09
        ;INSTRUCTION 'JMP 1$' PLACED HERE
        ;FOR LOOP ON ERROR
        ;CHECK FOR 0
        ;PASS
        ;ERROR
        ;-----
        .WORD -2
        ;CME UNUSED BIT TEST
        ;READ 1 FROM UNUSED CME BIT09
        .WORD 0
        NOP ;END OF TEST
        INC $TESTN ;INCREMENT TEST COUNTER
    
```

TEST # 42 - CME UNUSED BIT 10 TEST

1320

```

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

```

```

005462
005462 000004
005464 005474
005466 005474
005470 000000
005472 005502
005474
1321 005474 052737 002000 177744 40$:
1322 005502 000240 1$:
005504 000240 25$:
1323 005506 032737 002000 177744
1324 005514 001403
1325 005516 104413
005520 005516
1326
1327
1328 005522 000000
1329 005524 000240 10$:
005526 005237 001472

```

```

TEST42:
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
BIS #BIT10,CME ;WRITE 1 INTO BIT10
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIT #BIT10,CME ;CHECK FOR 0
BFO 10$ ;PASS
ERROR ;ERROR
:-----
.WORD -2
: CME UNUSED BIT TEST
:READ 1 FROM UNUSED CME BIT10
.WORD 0
NOP
INC $TESTN ;END OF TEST
;INCREMENT TEST COUNTER

```

1330

.SBTTL TEST # 43 - CME UNUSED BIT 11 TEST
 :.....
 :*TEST 43 CME UNUSED BIT 11 TEST
 :.....
 :*ST43:

005532	005532	00C004			SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON	
							:ERROR/LOOP ON TEST	
005534	005534	005544			.WORD	40\$:TEST START LOCATION	
005536	005536	005544			.WORD	1\$:LOOP ON ERROR START LOCATION	
005540	005540	000000			.WORD	0	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST	
005542	005542	005552			.WORD	25\$:LOOP ON ERROR END LOCATION	
005544				40\$:				
1331	005544	052737	004000	177744	1\$:	BIS	#BIT11,CME	:WRITE 1 INTO BIT11
1332	005552	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	005554	000240				NOP		:FOR LOOP ON ERROR
1333	005556	032737	004000	177744		BIT	#BIT11,CME	:CHECK FOR 0
1334	005564	001403				BEQ	10\$:PASS
1335	005566	104413				ERROR		:ERROR
								:-----
	005570	005566			.WORD	.-2		:CME UNUSED BIT TEST
1336								:READ 1 FROM UNUSED CME BIT11
1337								
1338	005572	000000			.WORD	0		:END OF TEST
1339	005574	000240			10\$:	NOP		:INCREMENT TEST COUNTER
	005576	005237	001472			INC	\$*ESTN	

1340

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

```
005602 000004
005604 005614
005606 005614
005610 000000
005612 005622
005614
1341 005614 052737 010000 177744 40$:
1342 005622 000240 1$: BIS #BIT12,CME ;WRITE 1 INTO BIT12
005624 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
1343 005626 032737 010000 177744 NOP ;FOR LOOP ON ERROR
1344 005634 001403 BIT #BIT12,CME ;CHECK FOR 0
1345 005636 104413 BEQ 10$ ;PASS
ERROR ;ERROR
;-----

005640 005636 .WORD -2 ;CME UNUSED BIT TEST
1346 ;READ 1 FROM UNUSED CME BIT12
1347
1348 005642 000000 .WORD 0
1349 005644 000240 10$: NOP ;END OF TEST
005646 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
```


1360

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

```
TST46:
      005722
      005722 000104          .SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      005724 005734          .WORD 40$         ;ERROR/LOOP ON TEST
      005726 005734          .WORD 1$         ;TEST START LOCATION
      005730 000000          .WORD 0          ;LOOP ON ERROR START LOCATION
      005732 005742          .WORD 25$        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      005734          40$:   .WORD 25$        ;LOOP ON ERROR END LOCATION
1361 005734 052737 040000 177744 1$:   BIS #BIT14,CME ;WRITE 1 INTO BIT14
1362 005742 000240          25$:   NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
      005744 000240          .NOP          ;FOR LOOP ON ERROR
1363 005746 032737 040000 177744 .BIT #BIT14,CME ;CHECK FOR 0
1364 005754 001403          BEQ 10$         ;PASS
1365 005756 104413          ERROR          ;ERROR
      005760 005756          .WORD -2        ;-----
1366          ;CME UNUSED BIT TEST
1367          ;READ 1 FROM UNUSED CME BIT14
1368 005762 000000          .WORD 0
1369 005764 000240          10$:  NOP
      005766 005237 001472 .INC $TESTN     ;END OF TEST
      ;INCREMENT TEST COUNTER
```

1375

```

.SBTTL TEST # 47 - CACHE CONTROL REGISTER BYTE TESTS (CCR)
*****
*TEST 47      CACHE CONTROL REGISTER BYTE TESTS (CCR)
*   REGISTER BYTE SELECTION LOGIC TEST
*   WRITE ONE INTO LOW BYTE WRITE ZERO INTO HIGH BYTE
*   VERIFY THAT LOW BYTE DATA IS NOT EFFECTED BY WRITE TO HIGH BYTE
*****
TST47:
    
```

005772	005772	000004				SCPCND		:SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
								:ERROR/LOOP ON TEST
005774	006004					.WORD	40\$:TEST START LOCATION
005776	006004					.WORD	1\$:LOOP ON ERROR START LOCATION
006000	000000					.WORD	0	:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
006002	006020					.WORD	25\$:LOOP ON ERROR END LOCATION
006004					40\$:			
1376	006004	152737	000004	177746	1\$:	BISB	#BIT02,CCR	:WRITE 1 INTO CONTROL REGISTER BIT02
1377	006012	142737	000004	177747		BICB	#BIT02,CCR+1	:WRITE 0 INTO CONTROL REGISTER BIT10
1378	006020	000240			25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	006022	000240				NOP		:FOR LOOP ON ERROR
1379	006024	032737	000004	177746		BIT	#BIT02,CCR	:CHECK FOR 1
1380	006032	001003				BNE	10\$:PASS
1381	006034	104413				ERROR		:ERROR
								:-----
	006036	006034				.WORD	.-2	:CACHE CONTROL REGISTER BYTE TESTS
1382								:WROTE ONE INTO LOW BYTE BIT02
1383								:WROTE ZERO INTO HIGH BYTE BIT10
1384								:READ ZERO FROM BIT02
1385								
1386	006040	000000				.WORD	0	
1387	006042	000240			10\$:	NOP		:END OF TEST
	006044	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

1392

```
.SBTTL TEST # 50 - SET TOP BYTE, CLEAR LOW BYTE OF CCR
:*****
:*TEST 50 SET TOP BYTE, CLEAR LOW BYTE OF CCR
:* WRITE ZERO INTO HIGH BYTE WRITE ONE INTO LOW BYTE
:* VERIFY HIGH BYTE NOT EFFECTED BY WRITE INTO LOW BYTE
:*****
```

```
TST50:
        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
        .WORD 40$
        .WORD 1$
        .WORD 0
        .WORD 25$
1393 006050 000004           40$: BICB #BIT02,CCR+1 :WRITE 0 INTO CONTROL REGISTER BIT10
1394 006052 006062           1$: BISB #BIT02,CCR :WRITE 1 INTO CONTROL REGISTER BIT02
1395 006054 006062           :NOP :INSTRUCTION 'JMP 1$' PLACED HERE
1396 006056 000000           :NOP :FOR LOOP ON ERROR
1397 006060 006076           25$: BIT #BIT10,CCR :CHECK FOR 0 BIT10
1398 006100 000240           :BEQ 10$ :PASS
1399 006110 001403           :ERROR :ERROR
1400 006112 104413           :-----
1401 006114 006112           .WORD -2 :CACHE CONTROL REGISTER BYTE TESTS
1402 : : :WROTE ZERO INTO HIGH BYTE BIT10
1403 : : :WROTE ONE INTO LOW BYTE BIT02
1404 : : :READ ZERO FROM BIT02 OR READ ONE FROM BIT10
        .WORD 0
        .WORD 10$ : : :END OF TEST
        INC $TESTN : : :INCREMENT TEST COUNTER
```


1408

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

```
TST51:
006126          SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
006126 000004          ;ERROR/LOOP ON TEST
006130 006140          .WORD 40$ ;TEST START LOCATION
006132 006140          .WORD 1$ ;LOOP ON ERROR START LOCATION
006134 000000          .WORD 0  ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
006136 006152          .WORD 25$ ;LOOP ON ERROR END LOCATION
006140          40$:
1409 006140 042737 000001 177750 1$: BIC #BIT00,CMR ;WRITE 0 INTO CMR BIT00
1410 006146 013700 177750          MOV CMR,RO ;SAVE CONTENTS OF CMR
1411 006152 000240          25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
006154 000240          NOP ;FOR LOOP ON ERROR
1412 006156 005037 177750          CLR CMR ;CLR MAINT
1413 006162 032700 000001          BIT #BIT00,RO ;CHECK FOR 0 IN BIT00
1414 006166 001403          BEQ 10$ ;PASS
1415 006170 104413          ERROR ;ERROR
;-----
006172 006170          .WORD -2 ;MAINTENANCE REGISTER DATA TEST
1416          ;WROTE 0 INTO CMR BIT00; READ 1
1417
1418 006174 000000          .WORD 0
1419 006176 000240          10$: NOP ;END OF TEST
006200 005237 001472          INC $TESTN ;INCREMENT TEST COUNTER
```

1427

```

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

```

```

006204 000004
006206 006216
006210 006216
006212 000000
006214 006230
006216
1424 006216 052737 000001 177750
1425 006224 013700 177750
1426 006230 000240
006232 000240
1427 006234 005037 177750
1428 006240 032700 000001
1429 006244 001003
1430 006246 104413

006250 006246
1431
1432
1433 006252 000000
1434 006254 000240
006256 005237 001472

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

.WORD -2
:MAINTENANCE REGISTER DATA TEST
:WROTE 1 INTO CMR BIT00; READ 0

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

```

1438

```
.SBITL TEST # 53 - TEST BIT 1 OF CMR
:*****
:*TEST 53 TEST BIT 1 OF CMR
:* VERIFY CMR BIT01(MODO) CAN BE WRITTEN AS A 0.
:*****
```

```
TST53:
006262 000004
006264 006274
006266 006274
006270 000000
006272 006306
006274
1439 006274 042737 000002 177750 40$:
1440 006302 013700 177750 1$:
1441 006306 000240 25$:
006310 000240
1442 006312 005037 177750
1443 006316 032700 000002
1444 006322 001403
1445 006324 104413

006326 006324
-446
1447
1448 006330 000000
1449 006332 000240 10$:
006334 005237 001472

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

:WRITE 0 INTO CMR BIT01
:SAVE CONTENTS OF CMR
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:CLR MAINT
:CHECK FOR 0 IN BIT01
:PASS
:ERROR
:-----

:MAINTENANCE REGISTER DATA TEST
:WROTE 0 INTO CMR BIT01; READ 1

:END OF TEST
:INCREMENT TEST COUNTER

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

BIC #BIT01,CMR
MOV CMR,R0
NOP
NOP
CLR CMR
BIT #BIT01,R0
BEQ 10$
ERROR

.WORD -2

.WORD 0
NOP
INC $TESTN
```


1468

```

.SBTTL TEST # 55 - TEST CMR BIT 3
:.....
:TEST 55 TEST CMR BIT 3
:* VERIFY CMR BIT03(AM) CAN B. WRITTEN AS A 0.
:.....
TST55:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        ;TEST START LOCATION
        ;LOOP ON ERROR START LOCATION
        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        ;LOOP ON ERROR END LOCATION

        .WORD 40$
        .WORD 1$
        .WORD 0
        .WORD 25$
40$:    MOV     #-1,CHR           ;ALL 1'S TO AMR
        MOVB  #3/4,CMR+1
        CLRB  CMR
        MOV   CMR,R0
25$:    NOP
        NOP
        CLR   CMR
        BIT   #BIT03,R0
        BEQ  10$
        ERROR
        ;-----

        .WORD  -2
1478    ;MAINTENANCE REGISTER DATA TEST
1479    ;WROTE 0 INTO CMR BIT03; READ 1

        .WORD  0
1480    NOP
1481    INC   $TESTN
10$:    ;END OF TEST
        ;INCREMENT TEST COUNTER
    
```

```

006416 000004
006420 006430
006422 006430
006424 000000
006426 006454
006430
1469 006430 012737 177777 177752
1470 006436 112737 000374 177751
1471 006444 105037 177750
1472 006450 013700 177750
1473 006454 000240
        006456 000240
1474 006460 005037 177750
1475 006464 032700 000010
1476 006470 001403
1477 006472 104413

006474 006472
1478
1479
1480 006476 000000
1481 006500 000240
        006502 005237 001472
    
```

1485

.SBTTL TEST # 56 - TEST CMR BIT 3

```

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

```

```

006506      000004
006510      006520
006512      006520
006514      000000
006516      006552
006520
1486 006520 012737 177777 177752
1487 006526 112737 000374 177751
1488 006534 105037 177750
1489 006540 112737 000010 177750
1490 006546 013700 177750
1491 006552 000240
006554 000240
1492 006556 005037 177750
1493 006562 032700 000010
1494 006566 001003
495 006570 104413

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

40$:
1$:  MOV      #-1,CHR
    MOVB    #374,CMR+1
    CLRB   CMR
    MOVB    #AM,CMR
    MOV     CMR,RO
25$:  NOP
    NOP
    CLR     CMR
    BIT     #BIT03,RO
    BNE    10$
    ERROR
:PRECONDITION AM BIT TO 0
:WRITE 1 INTO AM BIT
:SAVE CONTENTS OF CMR
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:CLR MAINT
:CHECK FOR 1 IN BIT03
:PASS
:ERROR
:-----

006572 006570
1496
1497
1498 006574 000000
1499 006576 000240
006600 005237 001472

.WORD  .-2
:MAINTENANCE REGISTER DATA TEST
:WROTE 1 INTO CMR BIT03; READ 0

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

```

```

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

```

```

006604
006604 000004
006606 006616
006610 006616
006612 000000
006614 006624
006616
1505 006616 052737 000040 177750 40$:
1506 006624 000240 1$:
006626 000240 25$:
1507 006630 032737 000040 177750
1508 006636 001403
1509 006640 104413
006642 006640
1510
1511
1512 006644 000000
1513 006646 000240 10$:
006650 005237 001472

```

```

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

BIS #BIT05,CMR ;WRITE 1 INTO BIT05
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIT #BIT05,CMR ;CHECK FOR 0
BEQ 10$ ;PASS
ERROR ;ERROR
;-----

.WORD -2
;CMR UNUSED BIT TEST
;READ 1 FROM UNUSED CMR BIT05

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

```

TEST # 60 - TEST UNUSED BIT 6 IN THE CMR

1114

```

.SBTTL TEST # 60 - TEST UNUSED BIT 6 IN THE CMR
.....
:TEST 60 TEST UNUSED BIT 6 IN THE CMR
.....
:TEST60:

```

```

006654 000004
006656 006666
006660 006666
006662 000000
006664 006674
006666
1515 006666 052737 000100 177750 40$:
1516 006674 000240 1$:
006676 000240 25$:
1517 006700 032737 000100 177750
1518 006706 001403
1519 006710 104413
006712 006710
1520
1521
1522 006714 000000
1523 006716 000240 10$:
006720 005237 001472
INC $TESTN

```

```

:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
:WRITE 1 INTO BIT06
:INSTRUCTION 'JMP 1$' PLACED HERE
:FOR LOOP ON ERROR
:CHECK FOR 0
:PASS
:ERROR
:-----
:CMR UNUSED BIT TEST
:READ 1 FROM UNUSED CMR BIT02
:END OF TEST
:INCREMENT TEST COUNTER

```


1524

..SBTTL TEST # 61 - TEST UNUSED BIT 7 IN THE CMR
 ..:.....
 ..TEST 61 TEST UNUSED BIT 7 IN THE CMR
 ..:.....

```

006724
006724 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
006726 006736          .WORD 40$          ;LOOP ON ERROR START LOCATION
006730 006736          .WORD 1$          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
006732 000000          .WORD 0           ;LOOP ON ERROR END LOCATION
006734 006744          .WORD 25$
006736
1525 006736 052737 000200 177750 40$: BIS #BIT07,CMR ;WRITE 1 INTO BIT07
1526 006744 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
                                ;FOR LOOP ON ERROR
006746 000240          NOP ;CHECK FOR 0
1527 006750 032737 000200 177750 BIT #BIT07,CMR ;PASS
1528 006756 001403          BEQ 10$          ;ERROR
1529 006760 104413          ERROR ;-----
                                ;-----
006762 006760          .WORD -2          ;CMR UNUSED BIT TEST
1530
1531
1532 006764 000000          .WORD 0           ;READ 1 FROM UNUSED CMR BIT03
1533 006766 000240 10$: NOP ;END OF TEST
006770 005237 001472          INC $TESTN        ;INCREMENT TEST COUNTER
    
```

1540

.SBTTL TEST # 62 - TEST AMR

```

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

```

006774
006774 000004

006776 007006
007000 007006
007002 000000
007004 007046
007006

1541 007006 012737 177777 177752
1542 007014 112737 000374 177751
1543 007022 105037 177750
1544 007026 005037 177752
1545 007032 105037 177751
1546 007036 005737 000000
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558 007042 013703 177750
1559 007046 000240
      007050 000240
1560 007052 032703 000010
1561 007056 001003
1562 007060 104413

      007062 007060

1563
1564
1565
1566
1567 007064 000000
1568 007066 000240
      007070 005237 001472

1569
1570
1571
1572
1573
1574 007074 012737 077406 172300
1575 007102 012737 077406 172302
1576 007110 012737 077406 172304
1577 007116 012737 077406 172306
    
```

```

TST62:
      SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      ;ERROR/LOOP ON TEST
      ;TEST START LOCATION
      .WORD 40$ ;LOOP ON ERROR START LOCATION
      .WORD 1$ ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0 ;LOOP ON ERROR END LOCATION
      .WORD 25$

40$:
1$: MOV #-1,CHR ;ALL 1'S TO AMR
     MOVB #374,CMR+1
     CLRB CMR ;PRECONDITION AM BIT TO 0
     CLR CHR ;ALL 0'S TO AMR<21:0>
     CLRB CMR+1
     TST 0 ;PLACE ALL 0'S ON CA<21:0>.HOWEVER,
           ;THIS IS NOT WHEN THE AM BIT IS SET:
           ;WHEN PAX ADDRESS LINES ARE NOT BEING
           ;ACCESSED BY THE CPU, THE CACHE DEFAULTS
           ;TO SELECTING MA<21:0> ADDRESS LINES.
           ;IN THIS SITUATION, MA<21:0> DEFAULTS
           ;TO ALL 1'S THEREBY PLACING ALL 0'S
           ;ON CA<21:0>. THEREFORE, FOLLOWING THE LOADING
           ;OF ALL 0'S INTO AMR<21:0>,AND BEFORE THE
           ;'TST 0' INSTRUCTION, THE AM BIT SHOULD
           ;BE SET DUE TO MATCH BETWEEN AMR<21:0> AND CA<21:0>
           ; ADDRESS LINES.
           ;SAVE AM BIT RESULT IN CMR
25$: MOV CMR,R3 ;INSTRUCTION 'JMP 1$' PLACED HERE
     NOP ;FOR LOOP ON ERROR
     BIT #AM,R3 ;AM BIT SHOULD READ 1 INDICATING MATCH
     BNE 10$ ;PASS
     ERROR ;ERROR
     ;-----

      .WORD -2 ;AMR TESTS
           ;AMR BIT DID NOT READ 1 INDICATING
           ;A MATCH OF ALL 0'S BETWEEN MA TO CA<21:0>
           ;ADDRESS LINES AND AMR<21:0> DATA
           ;ERROR PRINT TERMIN.
10$: .WORD 0 ;END OF TEST
     NOP ;INCREMENT TEST COUNTER
     INC $TESTN ;INCREMENT TEST COUNTER
     .SBTTL MEMORY MANAGEMENT AND UNIBUS MAP REGISTERS SETUP

*****
* MEMORY MANAGEMENT SETUP
*****
MAGPRE: MOV #77406,KPDR0 ;ALLOW ALL ACCESS TO KERNEL PAGE 0
        MOV #77406,KPDR1 ;ALLOW ALL ACCESS TO KERNEL PAGE 1
        MOV #77406,KPDR2 ;ALLOW ALL ACCESS TO KERNEL PAGE 2
        MOV #77406,KPDR3 ;ALLOW ALL ACCESS TO KERNEL PAGE 3
    
```

```

1578 007124 012737 077406 172310      MOV      #77406,KPDR4      ;ALLOW ALL ACCESS TO KERNEL PAGE 4
1579 007132 012737 077406 172312      MOV      #77406,KPDR5      ;ALLOW ALL ACCESS TO KERNEL PAGE 5
1580 007140 012737 077406 172314      MOV      #77406,KPDR6      ;ALLOW ALL ACCESS TO KERNEL PAGE 6
1581 007146 012737 077406 172316      MOV      #77406,KPDR7      ;ALLOW ALL ACCESS TO KERNEL PAGE 7
1582 007154 005037 172340      CLR      KPAR0             ;MAP PAGE 0 FOR 0-4K
1583 007160 012737 000200 172342      MOV      #200,KPAR1        ;MAP PAGE 1 FOR 4-8K
1584 007166 012737 000400 172344      MOV      #400,KPAR2        ;MAP PAGE 2 FOR 8-12K
1585 007174 012737 000600 172346      MOV      #600,KPAR3        ;MAP PAGE 3 FOR 12-16K
1586 007202 012737 177600 172356      MOV      #177600,KPAR7     ;MAP PAGE 7 FOR 124-128K

```

```

1587
1588      ;*****
1589      ;*      UNIBUS MAP REGISTERS SETUP
1590      ;*****

```

```

1591 007210 012737 000000 170200      MOV      #0,UMPR00         ;MAP REGISTER SET 0 FOR 0-4K
1592 007216 012737 000000 170202      MOV      #0,UMPR01
1593 007224 012737 020000 170204      MOV      #20000,UMPR02     ;MAP REGISTER SET 1 FOR 4K-8K
1594 007232 012737 000000 170206      MOV      #0,UMPR03
1595 007240 012737 040000 170210      MOV      #40000,UMPR04     ;MAP REGISTER SET 2 FOR 8K-12K
1596 007246 012737 000000 170212      MOV      #0,UMPR05
1597 007254 012737 060000 170214      MOV      #60000,UMPR06     ;MAP REGISTER SET 3 FOR 12K-16K
1598 007262 012737 000000 170216      MOV      #0,UMPR07

```

1605

```

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

```

007270
007270 000C74

007272 007302
007274 007302
007276 000000
007300 007326
007302
1606 007302 012737 177777 177752
1607 007310 112737 000374 177751
1608 007316 105037 177750
1609 007322 105737 177777
1610
1611 007326 000240
      007330 000240
1612 007332 032737 000010 177750
1613 007340 001003
1614 007342 104413

      007344 007342
1615
1616
1617
1618
1619 007346 000000
1620 007350 000240
      007352 005237 001472

TST63:
      SCPCND          ;SCOPE CONDITIONS.GO SET UP FOR LOOP ON
                      ;ERROR/LOOP ON TEST
      .WORD 40$       ;TEST START LOCATION
      .WORD 1$        ;LOOP ON ERROR START LOCATION
      .WORD 0         ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 25$       ;LOOP ON ERROR END LOCATION

40$:
      MOV #1,CHR      ;LOAD AMR<15:0> WITH 1'S FROM CHR<15:0>
1$:   MOVB #374,CMR+1 ;LOAD AMR<21:16> ALL 1'S
      CLRB CMR        ;PRECONDITION AM BIT TO 0
      TSTB 177777    ;PUT ALL 1'S ON MA AND PA ADDRESS LINES
                      ;MA WILL BE SELECTED
25$:  NOP            ;INSTRUCTION 'JMP 1$' PLACED HERE
      NOP            ;FOR LOOP ON ERROR
      BIT #AM,CMR    ;AM BIT SHOULD READ 1 INDICATING MATCH
      BNE 10$        ;PASS
      ERROR         ;ERROR
                      ;-----

      .WORD -2

                      ;AMR TESTS
                      ;AMR BIT DID NOT READ 1 INDICATING
                      ;A MATCH OF ALL 1'S BETWEEN CA<21:0>
                      ;ADDRESS LINES AND AMR<21:0> DATA
                      ;ERROR PRINT TERMIN.
10$:  .WORD 0
      NOP            ;END OF TEST
      INC $TESTN     ;INCREMENT TEST COUNTER
  
```

1627

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

007356
 007356 000004

007360 007370
 007362 060006
 007364 000000
 007366 060034
 007370 012737 001015 177746
 007376 004437 002424
 007402 007550

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

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

1628 007404 012737 000001 002066
 1629 007412 013737 002066 177752 1\$:
 1630 007420 105037 177751
 1631 007424 105037 177750
 1632 007430 005737 000000
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642 007434 013703 177750
 1643 007440 000240 25\$:
 007442 000240
 1644 007444 032703 000010
 1645 007450 001432
 1646 007452 005037 050466
 1647 007456 005037 050464
 1648 007462 013737 002066 050472
 1649
 1650 007470 013737 002066 050470
 1651 007476 012737 000017 002062
 1652 007504 006237 050470 3\$:
 1653 007510 042737 100000 050470
 1654 007516 005337 002062
 1655 007522 001370
 1656 007524 104413

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

007526 007524
 1657
 1658
 1659

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

1660	007530	050464		CA210
*661	007532	050470		AMR210
1662	007534	000000		.WORD 0
1663	007536	006337	002066	9\$: ASL CHRPAT
1664	007542	103401		BCS 10\$
1665	007544	000722		BR 1\$
*666	007546	000240		10\$: NOP
	007550	005237	001472	INC \$TESTN

```

:PRINT PATTERN USED FOR CACHE ADDRESS LINES CA<21:0>
:PRINT FLOATING 1 PATTERN USED FOR AMR<21:0> DATA

:NEXT FLOATING 1 PATTERN
:IF PHYSICAL ADDRESS 100000 DONE; THEN FINISHED
:IF NOT CONTINUE WITH NXT PATTERN
:END OF TEST
:INCREMENT TEST COUNTER
  
```

TEST OF AM

```

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

```

```

007554
007554 000004

007556 007566
007560 060006
007562 000000
007564 060034
007566 012737 001015 177746 40$
007574 004437 002424
007600 007730

```

```

TST65:      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
              ;ERROR/LOOP ON TEST
              ;TEST START LOCATION
              ;LOOP ON ERROR START LOCATION
              ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
              ;LOOP ON ERROR END LOCATION
              ;DISABLE CACHE
              ;LOCATE TEST CODE TO LOW CACHE SPACE
              ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

1674 007602 012737 000004 002064
1675 007610 113737 002064 177751 1$
1676 007616 005037 177752
1677 007622 105037 177750
1678 007626 005737 000000
1679
1680
1681 007632 013703 177750
1682 007636 000240 000240 25$
007640 U00240
1683 007642 032703 000010
1684 007646 001420
1685 007650 005037 050466
1686 007654 005037 050464
1687 007660 005037 050472
1688 007664 013737 002064 050470
1689 007672 006237 050470
1690 007676 104413

007700 007676

1691
1692
1693
1694 007702 050464
1695 007704 050470
1696 007706 000000
1697 007710 006337 002064 9$
1698 007714 032737 000400 002064
1699 007722 001001
1700 007724 000731
1701 007726 000240 001472 10$
007730 005237

```

```

MOV #4,CMRPAT ;SETUP 1ST PATTERN FOR CMR<21:16>
MOV#B CMRPAT,CMR+1 ;LOAD AMR<21:16> FROM CMR<15:0>
CLR CHR ;LOAD ALL 0'S TO AMR<15:0> FROM CHR<15:0>
CLR#B CMR ;PRECONDITION AM BIT TO 0
TST 0 ;SAVE CMR CONTENTS. BEFORE THE FETCH
;OF THIS INSTRUCTION, ALL 0'S WILL
;BE PLACED ON CA<21:0> LINES.
MOV CMR,R3 ;SAVE CMR CONTENTS
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
BIT #AM,R3 ;CHECK FOR 0.
BEQ 9$ ;PASS
CLR CA210+2 ;PREPARE CA210 FOR PRINTOUT
CLR CA210
CLR AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>
MOV CMRPAT,AMR210
ASR AMR210
ERROR ;ERROR
-----
;AMR TESTS
;AM BIT SHOULD HAVE READ 0 INDICATING A
;NO-MATCH CONDITION.
;PRINT PATTERN USED FOR CACHE ADDRESS LINES CA<21:0>
;PRINT FLOATING 1 PATTERN USED FOR AMR<21:0> DATA
;NEXT FLOATING 1 PATTERN
;IF PHYSICAL ADDRESS 10000000 DONE:FINISHED
;IF NOT CONTINUE WITH NXT PATTERN
;END OF TEST
;INCREMENT TEST COUNTER

```

... B CACHE ... MA TO CA LINES ARE SHORT TO EACH OTHER

.SBTTL TEST # 66 - VERIFY NO MA TO CA LINES ARE SHORT TO EACH OTHER

*TEST 66 VERIFY NO MA TO CA LINES ARE SHORT TO EACH OTHER
* MA<12:0> ADDRESS LINES FLOATING 0
* CA<12:0> ADDRESS LINES FLOATING 1
* AMR<12:0> FLOATING 1 PATTERN
* AM BIT READS 1

007734
007734 000004
007736 007746
007740 060020
007742 000000
007744 060114
007746 012737 001015 177746
007754 004437 002424
007760 010174

TST66:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
.WORD 40\$;TEST START LOCATION
.WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

1709 007762 012737 000001 050516 MOV #1,FLTPAT ;1ST FLOATING 1 PATTERN: 00001
1710 007770 012701 100001 MOV #100001,R1 ;LOAD VIRTUAL ADDRESS. SELECTS KPAR4
1711 ;AND SPECIFIES OFFSET FOR PHYSICAL ADDRESS.
1712 007774 012737 170000 172350 MOV #170000,KPAR4 ;MAP PAGE 4 FOR TOP 124K ADDRESSING.
1713 ;TOGETHER WITH VIRTUAL ADDRESS WILL
1714 ;PLACE 17000001 ON PA LINES AND
1715 ;00000001 ON MA LINES FOR 1ST FLOATING
1716 ;1 PATTERN.
1717 010002 012737 000001 177572 1\$: MOV #1,SRO ;ENABLE MEM. MNGMENT.
1718 010010 012737 000020 172516 MOV #20,SR3 ;ENABLE 22-BIT MAPPING
1719 010016 013737 050516 177752 MOV FLTPAT,CHR ;LOAD AMR WITH FLOATING 1 PATTERN
1720 010024 105037 177751 CLRB CMR+1
1721 010030 105037 177750 CLRB CMR ;PRECONDITION AM BIT TO 0
1722 010034 023727 050516 000001 CMP FLTPAT,#1 ;FOR 1ST PATTERN USE TSTB
1723 010042 001004 BNE 2\$
1724 010044 105711 TSTB (R1) ;
1725 010046 000240 NOP
1726 010050 000240 NOP
1727 010052 000403 BR 4\$
1728 010054 005711 2\$: TST (R1) ;
1729 010056 000240 NOP
1730 010060 000240 NOP
1731 010062 013703 177750 4\$: MOV CMR,R3
1732 010066 005037 177572 CLR SRO ;DISABLE MEM. MNGMNT.
1733 010072 005037 172516 CLR SR3
1734 010076 000240 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
010100 000240 NOP ;FOR LOOP ON ERROR
1735 010102 032703 000010 BIT #AM,R3 ;CHECK FOR 1
1736 010106 001017 BNE 9\$;PASS
1737 010110 013737 050516 050472 MOV FLTPAT,AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>
1738 ;FOR ERROR TYPEOUT
1739 010116 005037 050470 CLR AMR210
1740 010122 013737 050516 050466 MOV FLTPAT,CA210+2 ;PREPARE PATT. USED FOR CA<21:0 > FOR ERROR TYPE
1741 010130 005037 050464 CLR CA210
1742 010134 104413 ERROR ;ERROR


```

1743 010136 010134 .WORD .-2 ;-----
1744 ;AMR TESTS
1745 010140 050464 CA210 ;AM BIT DID NOT READ 1
1746 010142 050470 AMR210 ;PRINT FLOATING 1 PATTERN USED FOR (A<21:0>
1747 010144 000000 .WORD 0 ;PRINT FLOAT 1 PATRN. USED FOR AMR<21:0>
1748 010146 006337 050516 9$: ASL FLTPAT ;NXT PATTERN
1749 010152 032737 020000 050516 BIT #20000,FLTPAT ;IF PATTERN 10000 DONE · FINISHED
1750 010160 001004 BNE 10$ ;IF NOT NEXT PASS
1751 010162 006301 ASL R1
1752 010164 052701 100000 BIS #100000,R1
1753 010170 000704 BR 1$
1754 010172 000240 10$: NOP ;END OF TEST
010174 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
    
```

1761

..SBTTL TEST # 67 - AM FLOATING PATTERN TEST

```

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

```

010200      SCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
010200      000004      ;ERROR/LOOP ON TEST
010202      010212      ;TEST START LOCATION
010204      060036      ;LOOP ON ERROR START LOCATION
010206      000000      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
010210      060144      ;LOOP ON ERROR END LOCATION
010212      012737      001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
010220      004437      002424      ;SR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
010224      010552      ;.WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
  
```

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

```

1762 010226 012737 171000 172350 2$: MOV #171000,KPAR4 ;MAP PAGE 4 FOR TOP 124K ADDRESSING
1763                                     ;WILL ALSO SELECT UNIBUS MAP REGISTER
1764                                     ;SET #4
1765 010234 012737 000040 050516      MOV #40,FLTPAT ;SETUP 1ST PATTERN FOR UMPRO9
1766 010242 012737 020000 170220      MOV #20000,UMPRO8 ;SETUP 1ST PATTERN FOR UMPRO8
1767                                     ;ACCESSING TOP 124K,AND ENABLING
1768                                     ;UNIBUS MAP, THE 1ST ADDRESS WILL BE
1769                                     ;CONSTRUCTED THRU THE PA<21:0>
1770                                     ;LINES AS 17020000 AND AS 00020000
1771                                     ;THRU THE MA<21:0> LINES. DUE TO TOP 124K
1772                                     ;ADDRESSING CA<21:0> WILL SELECT THE MA LINES.
1773 010250 012737 000200 002064      MOV #200,CMRPAT ;SETUP 1ST PATTERN FOR CMR<15:10>
1774 010256 012737 020000 002066      MOV #20000,CHRPAT ;SETUP 1ST PATTERN FOR CHR<15:0>
1775 010264 012737 060126 000004 1$: MOV #3$-2$+60000,4 ;
1776 010272 012737 000340 000006      MOV #340,6 ;
1777 010300 113737 050517 170222      MOVFB FLTPAT+1,UMPRO9 ;LOAD UPPER BITS OF UNIBUS MAP REGISTER
1778 010306 113737 002065 177751      MOVFB CMRPAT+1,CMR+1 ;LOAD AMR<21:16> FROM CMR<15:10>
1779 010314 013737 002066 177752      MOV CHRPAT,CHR ;LOAD AMR<15:0> FROM CHR<15:0>
1780 010322 012737 000001 177572      MOV #1,SRO ;ENABLE MEM MNGMENT
1781 010330 012737 000060 172516      MOV #60,SR3 ;ENABLE UNIBUS MAP AND 22-BIT MAPPING
1782 010336 105037 177750      CLRB CMR ;PRECONDITION AM BIT WITH 0
1783 010342 005737 100000      TST 100000 ;TOP 124K ADDRESSING WILL BE DONE PLACING
1784                                     ;THE APPROPRIATE FLOATING 1 ADDRESS PATTERN
1785                                     ;ON CA<21:0>.HOWEVER, THIS IS NOT WHEN THE
1786                                     ;AM BIT IS SET: WHEN PAX ADDRESS LINES ARE
1787                                     ;NOT BEING ACCESSED BY THE CPU,THE CACHE
1788                                     ;DEFAULTS TO SELECTING MA<21:0> ADDRESS
1789                                     ;LINES. IN THIS SITUATION,MA<21:0> DEFAULTS TO
1790                                     ;WHATEVER ADDRESS PATTERN IS BEING SET UP
1791                                     ;VIA THE UNIBUS MAP.
1792                                     ;THEREFORE AFTER THE 'CLRB CMR' INSTRUCTION
1793                                     ;AND BEFORE 'TST 100000' THE AM BIT SHOULD
1794                                     ;BE SET
1795 010346 000240      NOP
1796 010350 000240      NOP
  
```


1-34

```

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

```

```

010556
010556 000004          SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
010560 010570          .WORD 40$          ;TEST START LOCATION
010562 060004          .JORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
010564 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
010566 060034          .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
010570 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
010576 004437 002424 .SR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
010602 010720          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

1839 010604 012701 000001          MOV #1,R1          ;R1 CONTAINS 1ST FLOATING 1 PATTERN: 000001
1840 010610 010137 177752          1$: MOV R1,CHR      ;LOAD AMR<15:0> FROM CHR<15:0>
1841 010614 105037 177751          CLRB CMR+1        ;LOAD AMR<21:16> FROM CMR<15:10>
1842 010620 105037 177750          CLRB CMR          ;PRECONDITION AM BIT TO 0
1843 010624 022701 000001          CMP #1,R1         ;IF PATTERN IS 000001 USE TSTB
1844 010630 001002          BNE 2$           ;
1845 010632 105711          TSTB (R1)        ;
1846 010634 000401          BR 3$            ;
1847 010636 005711          2$: TST (R1)     ;READ ADDRESS SPECIFIED IN R1
1848                                ;WHICH WILL PLACE FLOATING 1 PATTERN ON ADDRESS LINES
1849                                ;PA WILL BE SELECTED TO FEED CA LINES.
1850 010640          3$:
1851 010640 000240          25$: NOP           ;INSTRUCTION 'JMP 1$' PLACED HERE
010642 000240          NOP           ;FOR LOOP ON ERROR
1852 010644 032737 000010 177750   BIT #AM,CMR       ;CHECK FOR 1
1853 010652 001015          BNE 9$           ;PASS
1854 010654 010137 050472          MOV R1,AMR210+2 ;PREPARE PATTERN USED FOR AMR<21:0>
1855                                ;FOR ERROR TYPE.
1856 010660 005037 050470          CLR AMR210       ;
1857 010664 010137 050466          MOV R1,CA210+2  ;PREPARE PATTREN USED FOR CA<21:0>
1858 010670 005037 050464          CLR CA210       ;
1859 010674 104413          ERROR          ;ERROR
                                ;-----
010676 010674          .WORD -2       ;
1860                                ;AMR TESTS
1861                                ;AM BIT SHOULD HAVE READ 1 INDICATING A
1862                                ;MATCH CONDITION.
1863 010700 050464          CA210          ;PRINT FLOATING 1 PATTERN USED FOR
1864                                ;CACHE ADDRESS LINES CA<21:0>
1865 010702 050470          AMR210        ;PRINT PATTERN USED FOR AMR<21:0> DATA
1866 010704 000000          .WORD 0         ;
1867 010706 006301          9$: ASL R1       ;NEXT FLOATING 1 PATTERN
1868 010710 032701 100000          BIT #100000,R1  ;IS ADDRESS PATTERN 40000 DONE?
1869 010714 001735          BEQ 1$         ;NO; CONTINUE
1870 010716 000240          10$: NOP        ;END OF TEST

```

CRKACO 11-44 KK118 CACHE MACRO M1113 28-MAR-81 14:20 PAGE 78-1
TEST # 70 - VERIFY NO STUCK PA LINES OR SHORTED TO EACH OTHER

SEQUENCE 89

010720 005237 001472

INC STSTN

:INCREMENT TEST COUNTER

1876

..SBTTL TEST # 71 - AFTER EACH FLOATING 1 PAT. CHECK AM BIT READS 1

```

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

```

```

010724
010724 000004

010726 010736
010730 060022
010732 000000
010734 060116
010736 012737 001015 177746
010744 004437 002424
010750 011250

```

```

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

```

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

```

1877 010752 012737 000002 002064
1878 010760 012737 100000 002066
1879 010766 012737 001000 172350
1880
1881
1882
1883
1884
1885
1886 010774 012737 060104 000004
1887 011002 012737 000340 000006
1888 011010 113737 002064 177751
1889 011016 013737 002066 177752
1890 011024 012737 000001 177572
1891 011032 012737 000020 172516
1892 011040 105037 177750
1893 011044 005737 100000
1894
1895
1896 011050 000240
1897 011052 000240
1898 011054 000401
1899 011056 022626
1900 011060 005037 177572
1901 011064 005037 172516
1902 011070 000240
        011072 000240
1903 011074 032737 000010 177750
1904 011102 001044
1905 011104 012737 000006 000004
1906 011112 005037 000006
1907 011116 005037 050472
1908
1909 011122 005037 050466
1910 011126 013737 172350 050470
1911 011134 013737 172350 050464

```

```

2$: MOV #2,CMRPAT           :1ST PATTERN FOR CMR<15:10>
     MOV #100000,CHRPAT     :1ST PATTERN FOR CHR<15:0>
     MOV #1000,KPAR4        :SETUP 1ST PATRN. FOR PAGE ADDRESS FIELD
                                :KPAR4 CONTAINS THE FLOATING 1
                                :PATTERN AND REPRESENTS THE PAGE ADDRESS FIELD
                                :DATA USED BY MEM. MNGMNT. TO
                                :CONSTRUCT THE PHYSICAL ADDRESS.
                                :1000 IS THE 1ST FLOATING 1 PATTERN
                                :WHICH WILL BE CONSTRUCTED AS PHYS. ADDRESS 100000.
1$: MOV #6$-2$+60000,4     :ALLOW FOR NEX TRAP
     MOV #340,6
     MOVB CMRPAT,CMR+1      :LOAD AMR<21:16> FROM CMR<15:10>
     MOV CHRPAT,CHR         :LOAD AMR<15:0> FROM CHR<15:0>
     MOV #1,SRO             :ENABLE MEM. MNGMNT.
     MOV #20,SR3           :ENABLE 22-BIT MAPPING
     CLRB CMR               :PRECONDITION AM BIT TO 0
     TST 100000            :WILL CHOOSE KPAR4 FOR ADDRESSING.
                                :PHYSICAL ADDRESS WILL BE DETERMINED
                                :BY FLOATING PATTERN USED.

6$: CMP (SP)+,(SP)+       :NO TRAP
7$: CLR SRO                :ADJUST STACK
     CLR SR3               :DISABLE MEM. MNGMNT.

25$: NOP                   :INSTRUCTION 'JMP 1$' PLACED HERE
     NOP                   :FOR LOOP ON ERROR
     BIT #AM,CMR           :CHECK FOR 1
     BNE 9$                :PASS
     MOV #6,4
     CLR 6
     CLR AMR210+2         :PREPARE PATTERN USED FOR AMR<21:0>
                                :AND CA<21:0> FOR TYPEOUT
     CLR CA210+2
     MOV KPAR4,AMR210
     MOV KPAR4,CA210

```

1912	011142	012737	000011	002062		MOV	#9, LOOP	
1913	011150	006237	050470		5\$:	ASR	AMR210	
1914	011154	006237	050464			ASR	CA210	
1915	011160	042737	100000	050470		BIC	#100000, AMR210	
1916	011166	042737	100000	050464		BIC	#100000, CA210	
1917	011174	005337	002062			DEC	LOOP	
1918	011200	001363				BNE	5\$	
1919	011202	104413				ERROR		:ERROR
								:-----
	011204	011202				.WORD	.-2	
1920								:AMR TESTS
1921	011206	050464				CA210		:PRINT FLOAT. ADDRESS PATTERN USED
1922								:FOR CA<21:0>
1923	011210	050470				AMR210		:PRINT PATTERN USED FOR AMR<21:0>
1924	011212	000000				.WORD	0	
1925	011214	006337	172350		9\$:	ASL	KPAR4	:NEXT FLOATING 1 PATTERN
1926	011220	103405				BCS	8\$:IF PHYSICAL ADDRESS 10000000 DONE; FINISHED
1927	011222	006337	002064			ASL	CMRPAT	:NEXT CMR PATTERN
1928	011226	006337	002066			ASL	CHRPAT	:NEXT CHR PATTERN
1929	011232	000660				BR	1\$:CONTINUE
1930	011234	012737	000006	000004	8\$:	MOV	#6, 4	:RESTORE VECTORS
1931	011242	005037	000006			CLR	6	
1932	011246	000240			10\$:	NOP		:END OF TEST
	011250	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

1936

.SBTTL TEST # 72 - LOADING TAG STORE FROM ADDRESS MATCH REGISTERS

 :*TEST 72 LOADING TAG STORE FROM ADDRESS MATCH REGISTERS
 :* ALL 0'S TO TAG STORE ADDRESS LOCATION 0000.

011254										
011254	000004									TST72:
						SCPCND				:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
										:ERROR/LOOP ON TEST
011256	011266					.WORD	40\$:TEST START LOCATION
011260	070000					.WORD	1\$-40\$+67764			:LOOP ON ERROR START LOCATION
011262	000000					.WORD	0			:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
011264	070042					.WORD	25\$-40\$+67764			:LOOP ON ERROR END LOCATION
011266	012737	001015	177746	40\$:		MOV	#OFF,CCR			:DISABLE CACHE
011274	004437	002452				JSR	R4,RELCTH			:LOCATE TEST CODE TO HIGH CACHE SPACE
011300	011436					.WORD	10\$+2			:ADDRESS OF START OF NEXT TEST

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

1937	011302	005037	177752	1\$:	CLR	CHR				:LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
1938	011306	112737	000003	177750	MOV	#HODO+TDAR,CMR				:ALLOWS CACHE TAG FIELD BITS TO BE
1939										:WRITTEN TO CHR<15:07> ONLY DURING
1940										:THE DESTINATION MEMORY ACCESS
1941										:OF AN INSTRUCTION
1942										:ENABLE CACHE TAG FIELD TO BE WRITTEN
1943										:INTO FROM AMR<8:0>
1944	011314	012737	000015	177746	MOV	#15,CCR				:NO UCB SO AS TO WRITE ENABLE TAG STORE
1945	011322	005737	040000		TST	40000				
1946	011326	005737	060000		TST	60000				:WRITE INTO TAG STORE
1947	011332	005737	060000		TST	60000				:WRITE TAG FIELD DATA FROM CACHE ADDRESS
1948										:LOCATION 0000 INTO CHR.
1949	011336	013737	177752	050474	MOV	CHR,CHR157				:SAVE CHR DATA
1950	011344	000240		25\$:	NOP					:INSTRUCTION 'JMP 1\$' PLACED HERE
	011346	000240			NOP					:FOR LOOP ON ERROR
1951	011350	105037	177750		CLRB	CMR				:DISABLE MAINTENANCE MODE
1952	011354	012737	001015	177746	MOV	#OFF,CCR				
1953	011362	042737	000177	050474	BIC	#177,CHR157				:PREPARE CHR157 FOR ERROR CHECK
1954	011370	005737	050474		TST	CHR157				:BITS <15:07> SHOULD BE ALL 0'S
1955	011374	001417			BEQ	10\$:PASS
1956	011376	012737	000007	002062	MOV	#7,LOOP				:ERROR;PREPARE CHR157 FOR TYPEOUT
1957	011404	006237	050474	2\$:	ASR	CHR157				
1958	011410	042737	100000	050474	BIC	#100000,CHR157				
1959	011416	005337	002062		DEC	LOOP				
1960	011422	001370			BNE	2\$				
1961	011424	104413			ERROR					:ERROR
										:-----
	011426	011424			.WORD	.-2				
1962										:TAG STORE DATA TESTS
1963										:READING TAGD<21:13> THRU CHR<15:07>
1964										:DID NOT RESULT IN ALL 0'S.
1965	011430	050474			CHR157					:PRINT CHR<15:07>
1966	011432	000000			.WORD	0				
1967	011434	000240		10\$:	NOP					:END OF TEST
	011436	005237	001472		INC	\$TESTN				:INCREMENT TEST COUNTER

TEST # 73 - ALL 1'S TO TAG STORE ADDRESS LOCATION 0000

1971

```

.SBTTL TEST # 73 - ALL 1'S TO TAG STORE ADDRESS LOCATION 0000
:*****
:*TEST 73 ALL 1'S TO TAG STORE ADDRESS LOCATION 0000
:* ALL 1'S TO TAG STORE ADDRESS LOCATION 0000
:*****
TST73:

```

```

011442
011442 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
011444 011454          .WORD 40$          ;TEST START LOCATION
011446 070000          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
011450 000000          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
011452 070044          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
011454 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
011462 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
011466 011630          .WORD 10$+2       ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

1972 011470 012737 177777 177752 1$: MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
1973 011476 112737 000003 177750 MOVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
1974 ;WRITTEN TO CHR<15:07> ONLY DURING
1975 ;THE DESTINATION MEMORY ACCESS
1976 ;OF AN INSTRUCTION
1977 ;ENABLE CACHE TAG FIELD TO BE WRITTEN
1978 ;INTO FROM AMR<8:0>
1979 011504 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE
1980 011512 005737 040000 TST 40000
1981 011516 005737 060000 TST 60000 ;WRITE INTO TAG STORE
1982 011522 005737 060000 TST 60000 ;WRITE TAG FIELD DATA FROM CACHE ADDRESS
1983 ;LOCATION 0000 INTO CHR.
1984 011526 013737 177752 050474 MOV CHR,CHR157 ;SAVE CHR DATA
1985 011534 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
011536 000240 NOP ;FOR LOOP ON ERROR
1986 011540 105037 177750 CLRB CMR ;DISABLE MAINTENANCE MODE
1987 011544 012737 001015 177746 MOV #OFF,CCR
1988 011552 042737 000177 050474 BIC #177,CHR157
1989 011560 022737 177600 050474 CMP #177600,CHR157 ;BITS <15:07> SHOULD BE ALL 1'S
1990 011566 001417 BEQ 10$ ;PASS
1991 011570 012737 000007 002062 MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
1992 011576 006237 050474 2$: ASR CHR157
1993 011602 042737 100000 050474 BIC #100000,CHR157
1994 011610 005337 002062 DEC LOOP
1995 011614 001370 BNE 2$
1996 011616 104413 ERROR ;ERROR
;-----
011620 011616 .WORD -2
1997 ;TAG STORE DATA TESTS
1998 ;READING TAGD<21:13> THRU CHR<15:07>
1999 ;DID NOT RESULT IN ALL 1'S.
2000 011622 050474 CHR157 ;PRINT CHR<15:07>
2001 011624 000000 .WORD 0
2002 011626 000240 10$: NOP ;END OF TEST
011630 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```

2006

```
.SBTTL TEST # 74 - FLOAT 1 ACROSS 0'S TO TAG STORE ADRS LOC 0
*****
*TEST 74      FLOAT 1 ACROSS 0'S TO TAG STORE ADRS LOC 0
*      FLOAT 1 ACROSS 0'S TO TAG STORE ADDRESS LOCATION 0000
*****
```

```
TST74:
        SPCOND          ;SCOPE CONDITIONS GO SET UP FOR LOOP ON
        .WORD 40$      ;ERROR/LOOP ON TEST
        .WORD 1$-40$+67764 ;TEST START LOCATION
        .WORD 0        ;LOOP ON ERROR START LOCATION
        .WORD 25$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 40$:    MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
        JSR R4,RELCTH ;DISABLE CACHE
        .WORD 10$+2   ;LOCATE TEST CODE TO HIGH CACHE SPACE
        .WORD 10$+2   ;ADDRESS OF START OF NEXT TEST
```

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

```
2007 011662 012737 0C0001 050500      MOV #1,CHR80 ;1ST FLOATING 1 PATTERN:001
2008 011670 012737 000015 177746      MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2009 011676 013737 050500 177752      MOV CHR80,CHR ;LOAD AMR<8:0> VIA CHR<8:0> WITH
2010                                     ;FLOATING 1 PATTERN
2011 011704 112737 000003 177750      MOV# #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
2012                                     ;WRITTEN TO CHR<15:07> ONLY DURING
2013                                     ;THE DESTINATION MEMORY ACCESS
2014                                     ;OF AN INSTRUCTION
2015                                     ;ENABLE CACHE TAG FIELD TO BE WRITTEN
2016                                     ;INTO FROM AMR<8:0>
2017 011712 005737 040000      TST 40000
2018 011716 005737 060000      TST 60000 ;WRITE INTO TAG STORE
2019 011722 005737 060000      TST 60000 ;WRITE TAG FIELD DATA FROM CACHE ADDRESS
2020                                     ;LOCATION 0000 INTO CHR.
2021 011726 013737 177752 050474      MOV CHR,CHR157 ;SAVE CHR DATA
2022 011734 000240      25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      011736 000240      NOP ;FOR LOOP ON ERROR
2023 011740 105037 177750      CLR# CMR ;DISABLE MAINTENANCE MODE
2024 011744 012737 001015 177746      MOV #OFF,CCR ;DISABLE CACHE
2025 011752 012737 000007 002062      MOV #7,LOOP ;PREPARE CHR157 FOR COMPARISON
2026 011760 006237 050474      3$: ASR CHR157
2027 011764 042737 100000 050474      BIC #100000,CHR157
2028 011772 005337 002062      DEC LOOP
2029 011776 001370      BNE 3$
2030 012000 023737 050500 050474      CMP CHR80,CHR157 ;CHECK FOR CORRECT PATTERN
2031 012006 001405      BEQ 9$ ;PASS
2032 012010 104413      ERROR ;ERROR
      012012 012010      .WORD -2 ;-----
2033                                     ;TAG STORE DATA TESTS
2034                                     ;READING CHR<15:07> FOR TAGD<21:13>
2035                                     ;DID NOT RESULT IN CORRECT FLOATING
2036                                     ;1 PATTERN.
2037 012014 050474      CHR157 ;PRINT CHR<15:07>
2038 012016 050500      CHR80 ;PRINT FLOATING 1 PATTERN LOADED
2039                                     ;INTO CHR<8:0>
2040 012020 000000      .WORD 0
2041 012022 006337 050500      9$: ASL CHR80 ;NEXT PATTERN
```

2042	012026	032737	001000	050500	BIT	#1000.CHR80	:IF PATTERN 400 DONE;FINISHED
2043	012034	001715			BEQ	1\$:IF NOT, NEXT PASS
2044	012036	000240			10\$: NOP		:END OF TEST
	012040	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

SBTTL TEST # 75 - CHECK ALL LOW CACHE TAG STORE ADRS LOCS
 TEST 75 CHECK ALL LOW CACHE TAG STORE ADRS LOCS
 WRITE AND READ 0'S TO ALL LOW CACHE TAG STORE ADDRESS LOCATIONS

```

012044
012044 000004          SCPCND          :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                :ERROR/LOOP ON TEST
012046 012056          .WORD 40$          :TEST START LOCATION
012050 070014          .WORD 1$-40$+67764 :LOOP ON ERPOP START LOCATION
012052 000000          .WORD 0           :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
012054 070044          .WORD 25$-40$+67764 :LOOP ON ERROR END LOCATION
012056 012737 001015 177746 40$: MOV #OFF,CCR :DISABLE CACHE
012064 004437 002452 JSR R4,RELCTH :LOCATE TEST CODE TO HIGH CACHE SPACE
012070 012260          .WORD 10$+2       :ADDRESS OF START OF NEXT TEST
    
```

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

```

2049 012072 005037 177752          CLR CHR          :LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
2050 012076 012705 060000          MOV #60000,R5   :ADDRESS 60000 INTO R5
2051 012102 012703 040000          MOV #40000,R3   :ADDRESS 40000 INTO R3
2052 012106 012737 000015 177746 1$: MOV #15,CCR      :NO UCB SO AS TO WRITE ENABLE CACHE STORE
2053 012114 112737 000003 177750 MOVB #HODO+TDAR,CMR :ALLOWS CACHE TAG FIELD BITS TO BE
2054                                     :WRITTEN TO CHR<15:07> ONLY DURING
2055                                     :THE DESTINATION MEMORY ACCESS
2056                                     :OF AN INSTRUCTION
2057                                     :ENABLE CACHE TAG FIELD TO BE WRITTEN
2058                                     :INTO FROM AMR<8:0>
2059 012122 005713          TST (R3)        :
2060 012124 005715          TST (R5)        :WRITE INTO TAG STORE
2061 012126 005715          TST (R5)        :WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2062                                     :LOCATION SPECIFIED BY CA<12:1> IN R5.
2063 012130 013737 177752 050474 MOV CHR,CHR157  :SAVE CHR DATA
2064 012136 000240          NOP             :INSTRUCTION 'JMP 1$' PLACED HERE
2065 012140 000240          NOP             :FOR LOOP ON ERROR
2066 012142 010537 177750          CLRB CMR        :DISABLE MAINTENANCE MODE
2067 012146 012737 001015 177746 MOV #OFF,CCR     :DISABLE CACHE
2068 012154 042737 000177 050474 BIC #177,CHR157 :PREPARE CHR157 FOR ERROR CHECK
2069 012162 005737 050474          TST CHR157     :BITS <15:07> SHOULD BE ALL 0'S
2070 012166 001424          BEQ 9$         :PASS
2071 012170 010537 050466          MOV R5,CA210+2 :SAVE CACHE ADDRESS USED: CA<21:0>
2072 012174 005037 050464          CLR CA210
2073 012200 012737 000007 002062 MOV #7,LOOP      :ERROR;PREPARE CHR157 FOR TYPEOUT
2074 012206 006237 050474          ASR CHR157
2075 012212 042737 100000 050474 BIC #100000,CHR157
2076 012220 005337 002062          DEC LOOP
2077 012224 001370          BNE 4$
2078                                     :ERROR
2079                                     :-----
2080          012230 012226          .WORD -2
2081          CHR157
2082          CA210
2083                                     :TAG STORE DATA TESTS
2084                                     :READING TAGD<21:13> THRU CHR<15:07>
2085                                     :DID NOT RESULT IN ALL 0'S.
2086                                     :PRINT CHR<15:07>
2087                                     :PRINT CA<21:0> ADDRESS USED
2088                                     :BITS <12:1> IS THE CACHE TAG STORE ADDRESS
    
```

2084			
2085	012236	000000	
2086	012240	062705	000002
2087	012244	062703	000002
2088	012250	020527	070000
2089	012254	001314	
2090	012256	000240	
	012260	005237	001472

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

```

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

```

2095

```

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

```

012264
012264 000004          TST76:          SCPCND          :SCOPE CONDITIONS GO SET UP FOR LOOP ON
                                :ERROR/LOOP ON TEST
012266 012276          :.WORD 40$          :TEST START LOCATION
012270 070016          :.WORD 1$-40$+67764 :LOOP ON ERROR START LOCATION
012272 000000          :.WORD 0            :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
012274 070046          :.WORD 25$-40$+67764 :LOOP ON ERROR END LOCATION
012276 012737 001015 177746 40$: MOV #OFF,CCR          :DISABLE CACHE
012304 004437 002452          :JSR R4,RELCTH       :LOCATE TEST CODE TO HIGH CACHE SPACE
012310 012504          :.WORD 10$+2        :ADDRESS OF START OF NEXT TEST
  
```

```

:THE FOLLOWING LOCATIONS INCLUDING 0$
:ARE RELOCATED TO HI CACHE SPACE
  
```

```

2096 012312 012737 177777 177752          MOV #-1,CHR          :LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
2097 012320 012705 060000          MOV #60000,R5        :ADDRESS 60000 INTO R5
2098 012324 012703 040000          MOV #40000,R3        :ADDRESS 40000 INTO R3
2099 012330 012737 000015 177746 1$: MOV #15,CCR          :NO UCB SO AS TO WRITE ENABLE CACHE STORE
2100 012336 112737 000003 177750          MOV#B #HODO+TDAR,CMR :ALLOWS CACHE TAG FIELD BITS TO BE
2101          :WRITTEN TO CHR<15:07> ONLY DURING
2102          :THE DESTINATION MEMORY ACCESS
2103          :OF AN INSTRUCTION
2104          :ENABLE CACHE TAG FIELD TO BE WRITTEN
2105          :INTO FROM AMR<8:0>
2106 012344 005713          TST (R3)            :
2107 012346 005715          TST (R5)            :WRITE INTO TAG STORE
2108 012350 005715          TST (R5)            :WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
2109          :LOCATION SPECIFIED BY CA<12:1> IN R5
2110 012352 013737 177752 050474          MOV CHR,CHR157      :SAVE CHR DATA
2111 012360 000240          NOP                 :INSTRUCTION 'JMP 1$' PLACED HERE
2112 012364 105037 177750          NOP                 :FOR LOOP ON ERROR
2113 012370 012737 001015 177746          CLRB CMR            :DISABLE MAINTENANCE
2114 012376 042737 000177 050474          MOV #OFF,CCR        :
2115 012404 022737 177600 050474          BIC #177,CHR157     :
2116 012412 001424          CMP #177600,CHR157 :BITS <15:07> SHOULD BE ALL 1'S
2117 012414 010537 050466          BEQ 9$              :PASS
2118 012420 005037 050464          MOV R5,CA210+2     :SAVE CACHE ADDRESS USED: CA<21:0>
2119 012424 012737 000007 002062          CLR CA210          :
2120 012432 006237 050474          MOV #7,LOOP        :ERROR;PREPARE CHR157 FOR TYPEOUT
2121 012436 042737 100000 050474 4$: ASR CHR157          :
2122 012444 005337 002062          BIC #100000,CHR157 :
2123 012450 001370          DEC LOOP           :
2124 012452 104413          BNE 4$             :
2125          ERROR          :ERROR
2126          :-----
2127          :TAG STORE DATA TESTS
2128 012454 012452          :.WORD -2          :READING TAGD<21:13> THRU CHR<15:07>
2129 012460 050464          (CA21)            :DID NOT RESULT IN ALL 1'S.
2129 012460 050474          CHR157            :PRINT CACHE ADDRESS CA<21:0>
2129 012460 050474          :PRINT CHR<15:07>
  
```

2130	012462	000000	
2131	012464	062705	000002
2132	012470	062703	000002
2133	012474	020527	070000
2134	012500	001313	
2135	012502	000240	
	012504	005237	001472

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

:NEXT CACH LOCATION
 :HAVE ALL LOCATIONS BEEN DONE?
 :NO
 :END OF TEST
 :INCREMENT TEST COUNTER

2140

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

012510
012510 000004

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

012512 012522
012514 060014
012516 000000
012520 060044
012522 012737 001015 177746 40\$:
012530 004437 002424
012534 012724

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

2141 012536 005037 177752
2142 012542 012705 070000
2143 012546 012703 050000
2144 012552 012737 000015 177746 1\$:
2145 012560 112737 000003 177750
2146
2147
2148
2149
2150
2151 012566 005713
2152 012570 005715
2153 012572 005715
2154
2155 012574 013737 177752 050474
2156 012602 000240 25\$:
012604 000240
2157 012606 105037 177750
2158 012612 012737 001015 177746
2159 012620 042737 000177 050474
2160 012626 005737 050474
2161 012632 001424
2162 012634 010537 050466
2163 012640 005037 050464
2164 012644 012737 000007 002062
2165 012652 006237 050474 4\$:
2166 012656 042737 100000 050474
2167 012664 005337 002062
2168 012670 001370
2169 012672 104413

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

012674 012672
2170
2171
2172
2173 012676 050474
2174 012700 050464

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


```

2175
2176
2177 012702 000000
2178 012704 062705 000002
2179 012710 062703 000002
2180 012714 020527 100000
2181 012720 001314
2182 012722 000240
      012724 005237 001472

```

```

          .WORD 0
GS:      ADD #2,R5
          ADD #2,R3
          CMP R5,#100000
          BNE 1$
TC$:     NOP
          INC $TESTN

```

```

;BITS <12:1> IS THE CACHE TAG STORE ADDRESS
;LOCATION FAILURE
;NEXT CACHE STORE LOCATION
;HAVE ALL HI CACHE ADDRESS LOCATIONS BEEN CHECKED?
:NO
;END OF TEST
;INCREMENT TEST COUNTER

```

TEST CLEARING OF ALL HI CACHE TAG STORE LOCATIONS

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

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

012730
012730 000004

012732 012742
012734 060016
012736 000000
012740 060046
012742 012737 001015 177746 40\$
012750 004437 002424
012754 013150

TST100:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
ERROR/LOOP ON TEST
WORD 40\$;TEST START LOCATION
WORD 1\$-40\$+57764 ;LOOP ON ERROR START LOCATION
WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
WORD 25\$-40\$+57764 ;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

2188 012756 012737 177777 177752
2189 012764 012705 070000
2190 012770 012703 050000
2191 012774 012737 000015 177746 1\$
2192 013002 112737 000003 177750
2193
2194
2195
2196
2197
2198 013010 005713
2199 013012 005715
2200 013014 005715
2201
2202 013016 013737 177752 050474
2203 013024 000240 25\$
013026 000240
2204 013030 105037 177750
2205 013034 012737 001015 177746
2206 013042 042737 000177 050474
2207 013050 022737 177600 050474
2208 013056 001424
2209 013060 010537 050466
2210 013064 005037 050464
2211 013070 012737 000007 002062
2212 013076 006237 050474 4\$
2213 013102 042737 100000 050474
2214 013110 005337 002062
2215 013114 001370
2216 013116 104413

MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
MOV #70000,R5 ;ADDRESS 70000 INTO R5
MOV #50000,R3 ;ADDRESS 50000 INTO R3
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
MOVVB #HODO+TDAR,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
WRITTEN TO CHR<15:07> ONLY DURING
THE DESTINATION MEMORY ACCESS
OF AN INSTRUCTION
ENABLE CACHE TAG FIELD TO BE WRITTEN
INTO FROM AMR<8:0>
TST (R3)
TST (R5) ;WRITE INTO TAG STORE
TST (R5) ;WRITE TAG STORE DATA INTO CHR FROM CACHE ADDRESS
LOCATION SPECIFIED BY CA<12:1> IN R5
MOV CHR,CHR157 ;SAVE CHR DATA
NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINTENANCE
MOV #OFF,CCR
BIC #177,CHR157
CMP #177600,CHR157 ;BITS <15:07> SHOULD BE ALL 1'S
BEQ 9\$;PASS
MOV R5,CA210+2 ;SAVE CACHE ADDRESS USED: CA<21:0>
CLR CA210
MOV #7,LOOP ;ERROR:PREPARE CHR157 FOR TYPEOUT
ASR CHR157
BIC #100000,CHR157
DEC LOOP
BNF 4\$
ERROR ;ERROR

013120 013116
2217
2218
2219
2220 013122 050464
2221 013124 050474

.WORD -2
TAG STORE DATA TESTS
READING TAGD<21:13> THRU CHR<15:07>
DID NOT RESULT IN ALL 1'S.
PRINT CACHE ADDRESS CA<21:0>
PRINT CHR<15:07>

2222	013126	000000			.WORD	0	
2223	013130	062705	000002	9\$:	ADD	#2,R5	:NEXT CACH LOCATION
2224	013134	062703	000002		ADD	#2,R3	
2225	013140	020527	100000		CMP	R5,#100000	:HAVE ALL LOCATIONS BEEN DONE?
2226	013144	001313			BNE	1\$:NO
2227	013146	000240		10\$:	NOP		:END OF TEST
	013150	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

2270

```

.SBTTL TEST # 102 - WRITE FLOATING 1 ACROSS 0'S INTO TAG LOC 0
*****
*TEST 102 WRITE FLOATING 1 ACROSS 0'S INTO TAG LOC 0
* WRITE FLOATING 1 ACROSS 0'S INTO TAG STORE LOCATION 0000
* FROM CA<21:13> USING AVAILABLE MEMORY.
* PROCEDURE: STARTING AT 8K BOUNDARY(ADDR. 20000) CHECK
* FOR AVAILABLE FLOATING ADDRESS UP TO ADDR. 10000000
* WHEN THE FLOATING ADDRESS EXISTS PERFORM THE TEST.
*****
TST102:
  
```

```

013336
013336 000004

013340 013350
013342 070006
013344 000000
013346 070100
013350 012737 001015 177746
013356 004437 002452
013362 013650
  
```

```

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

```

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

```

2271 013364 012737 000200 172350 2$: MOV #200,KPAR4 ;KPAR4 CONTAINS THE FLOATING 1 PATTERN
2272 ;AND REPRESENTS THE THE PAGE ADDRESS FIELD
2273 ;DATA USED BY MEMORY MNGMNT. TO CONSTRUCT
2274 ;THE PHYSICAL ADDRESS. 200 IS THE 1ST
2275 ;FLOATING 1 PATTERN WHICH WILL BE CONSTRUCTED
2276 ;TO ADDRESS 20000.
2277 013372 012737 070240 000004 1$: MOV #7$-2$+70000,4 ;ALLOW FOR NEX TRAP
2278 013400 012737 000340 000006 MOV #340,6
2279 013406 112737 000002 177750 MOVB #HODO,CMR ;ALLOWS CACHE TAG STORE TO BE WRITTEN
2280 ;TO CHR<15:07> ONLY DURING THE DESTINATION
2281 ;MEMORY ACCESS OF AN INSTRUCTION.
2282 013414 012737 000001 177572 MOV #1,SRO ;ENABLE MEMORY MNGMNT.
2283 013422 012737 000020 172516 MOV #20,SR3 ;ENABLE 22-BIT MAPPING
2284 013430 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2285 013436 005737 040000 TST 40000
2286 013442 005737 100000 TST 100000 ;CHOOSES KPAR4 FOR ADDRESSING. PHYSICAL
2287 ;ADDRESS WILL BE DETERMINED BY FLOATING
2288 ;PATTERN USED IN KPAR4. TAG STORE WILL
2289 ;BE WRITTEN WITH DATA PLACED ON CA<21:13> ADDRESS LINES.
2290 013446 000240 NOP
2291 013450 000240 NOP ;NO TRAP
2292 013452 005737 100000 TST 100000 ;WRITE TAG STORE DATA FROM LOCATION
2293 ;0000 INTO CHR<15:07>.
2294 013456 013737 177752 050474 MOV CHR,CHR157 ;SAVE CHR INFO.
2295 013464 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
2296 013470 005037 177572 NOP ;FOR LOOP ON ERROR
2297 013474 005037 172516 CLR SRO ;DISABLE MEM MNGMENT.
2298 013500 012737 001015 177746 CLR SR3
2299 013506 105037 177750 MOV #OFF,CCR ;DISABLE CACHE
2300 013512 042737 000177 050474 CLRB CMR ;DISABLE MAINTENANCE
2301 013520 023737 172350 050474 BIC #177,CHR157
2302 013526 001437 CMP KPAR4,CHR157 ;IS THERE ERROR?
2303 013530 012737 000006 000004 BEQ 9$ ;PASS
MOV #6,4
  
```

```

2304 013536 005037 000006          CLR      6
2305 013542 013737 172350 050476    MOV      KPAR4,CA2113 ;SAVE PATTERN USED FOR CA<21:13>.
2306 013550 012737 000007 002062    MOV      #7,LOOP      ;PREPARE CHR157 AND CA2113 FOR ERROR PRINT
2307 013556 006237 050474          ASR      CHR157
2308 013562 006237 050476          ASR      CA2113
2309 013566 042737 100000 050474    BIC      #100000,CHR157
2310 013574 042737 100000 050476    BIC      #100000,CA2113
2311 013602 005337 002062          DEC      LOOP
2312 013606 001363          BNE      5$
2313 013610 104413          ERROR      ;ERROR
                                     ;-----
                                     .WORD      .-2
2314                                     ;TAG STORE TESTS
2315                                     ;READING CHR<15:07> FOR TAG DATA (TAGD <21:13>)
2316                                     ;DID NOT RESULT IN CORRECT ADDRESS PATTERN
2317                                     ;LOADED FROM CA<21:13>.
2318 013614 050474          CHR157    ;PRINT CHR<15:07>
2319 013616 050476          CA2113    ;PRINT ADDRESS PATTERN USED: CA<21:13>
2320 013620 000000          .WORD      0      ;PRINT TERMINATE
2321 013622 000401          BR        9$      ;NEXT PATTERN
2322 013624 022626          7$: CMP      (SP)+,(SP)+ ;RESTORE STACK DUE TO INTERRUPT
2323 013626 006337 172350          9$: ASL      KPAR4    ;NEXT PATTERN;IF PHYSICAL ADDRESS
                                     ;10000000 HAS BEEN DONE; FINISHED
2324
2325 013632 103257          BCC      1$
2326 013634 012737 000006 000004    MOV      #6,4      ;RESTORE VECTORS
2327 013642 005037 000006          CLR      6
2328 013646 000240          10$: NOP
                                     ;END OF TEST
                                     ;INCREMENT TEST COUNTER
2328 013650 005237 001472          INC      $TESTN

```

2339

..SBTTL TEST # 103 - FLOAT 1 ACROSS 0'S INTO TAG STORE ADRS LOC 0

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

013654
 013654 000004

TST103:

```

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

```

```

013656 013666
013660 070050
013662 000000
013664 070166
013666 012737 001015 177746 40$:
013674 004437 002452
013700 014426

```

```

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

```

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

```

2340 013702 132737 000200 001507 2$:
2341 013710 001405
2342 013712 032737 000200 001512
2343
2344 013720 001006
2345 013722 000555
2346 013724 012737 070352 000004 11$:
2347 013732 005737 177770
2348 013736 012737 001000 172350 5$:
2349
2350 013744 012737 000002 050516
2351
2352 013752 012737 070100 000004 1$:
2353 013760 012737 000001 177572
2354 013766 012737 000020 172516
2355 013774 005737 100000
2356 014000 000512
2357
2358 014002 022626 8$:
2359
2360 014004 013701 050516
2361 014010 005101
2362 014012 110137 177770
2363 014016 112737 000002 177750
2364
2365
2366 014024 012737 000015 177746
2367 014032 052737 000400 177770
2368 014040 005737 040000
2369 014044 005737 100000
2370

```

```

BITB #APTSIZE,$ENVM ;DOES APT SIZE?
BEQ 11$ ;NO ,GO AUTOSIZE
BIT #200,$USWR ;DOES APT INDICATE
;THAT RMI REGISTER IS PRESENT
BNE 5$ ;YES,USE IT
BR 4$ ;APT SAYS DO NOT PERFORM TEST
MOV #3$-2$+70000,4 ;AUTO-SIZE FOR RMI,PREPARE FOR TRAP
TST 177770 ;READ RMI
MOV #1000,KPAR4 ;SETUP MEM. MNG. PAGE 4 FOR FIRST FLOATING
;ADDRESS 100000
MOV #2,FLTPAT ;SETUP 1ST FLOATING PATTERN FOR RMI
;REG. CORRESPONDING TO ADDRESS 40000
MOV #8$-2$+70000,4 ;SETUP FOR NEX MEMORY
MOV #1,SRO ;ENABLE MEM.MNGMENT.
MOV #20,SR3 ;ENABLE 22 BIT MAPPING
TST 100000 ;SELECT PAGE 4. READ ADDRESS SPECIFIED BY KPAR4.
BR 9$ ;NO TRAP.MEMORY LOCATION EXISTS,SO DON'T
;BOTHER TESTING WITH RMI FOR THIS LOCATION
CMP (SP)+,(SP)+ ;TRAP HERE WHEN FLOATING ADDRESS
;LOCATION DOES NOT EXIST.USE RMI FOR TESTING
MOV FLTPAT,R1 ;PREPARE FLTPAT FOR LOADING INTO RMI
COM R1
MOVB R1,177770 ;LOAD RMI REGISTER
MOVB #HODO,CMR ;ALLOWS CACHE TAG STORE TO BE WRITTEN
;TO CHR<15:07> DURING THE DESTINATION
MOV #15,CCR ;MEMORY ACCESS OF AN INSTRUCTION ONLY
BIS #400,177770 ;NO UCB TO ENABLE TAG STORE WRITING
TST 40000 ;ENABLE RMI
TST 100000 ;SELECT PAGE 4 AND READ FLOATING ADDRESS
;SPECIFIED BY KPAR4. RMI WILL RESPOND

```


RRRACC 11-44 RR118 CACHE MACRO M1113 28-MAR-81 14:20 PAGE 89-2
TEST # '03 - FLOAT 1 ACROSS 0'S INTO TAG STORE ADRS LOC 0

SEQUENCE

0'4426 005237 00'4/2

INC STSTN

:INCREMENT TEST COUNTER

```

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

```

014432
014432 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
014434 014444          .WORD 40$          ;TEST START LOCATION
014436 070032          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
014440 000000          .WORD 0            ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
014442 070072          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
014444 012737 001015 177746 40$: MOV #OFF,CCP ;DISABLE CACHE
014452 004437 002452     JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
014456 014640          .WORD 10$+2        ;ADDRESS OF START OF NEXT TEST
    
```

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

```

2428 014460 012737 000002 050516     MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
2429 014466 012702 040000 2$: MOV #40000,R2 ;
2430 014472 012703 060000     MOV #60000,R3 ;
2431 014476 063702 050516     ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
2432 014502 063703 050516     ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
2433 014506 012713 177777     MOV #-1,(R3) ;ALL 1'S TO MAIN MEM. LOCATION
2434          ;SPECIFIED BY R3
2435 014512 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS TAG STORE BIT TO BE
2436          ;WRITTEN TO CHR<15:07> ONLY DURING THE
2437          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2438 014520 012737 000015 177746     MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2439 014526 005737 040000     TST 40000
2440 014532 005737 000000     TST 0 ;READ UPDATE; WRITE ALL 0'S INTO CACHE
2441          ;TAG STORE LOCATION 0000.
2442 014536 005712     TST (R2)
2443 014540 005713     TST (R3) ;READ UPDATE;WRITE BIT PATTERN 000000011
2444          ;INTO TAG STORE LOCATION SPECIFIED
2445          ;BY R3'S BITS 1 THRU 12: CA<12:1>
2446 014542 005737 060000     TST 60000 ;LOAD TAG STORE LOCATION 0000 INTO CHR
2447 014546 013701 177752     MOV CHR,R1 ;SAVE CHR CONTENTS
2448 014552 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
2449 014556 000240     NOP ;FOR LOOP ON ERROR
2450 014562 105037 177750     CLRB CMR ;DISABLE MAINT. MODE
2451 014570 012737 001015 177746     MOV #OFF,CCR ;DISABLE CACHE
2452 014574 042701 000177     BJC #177,R1 ;INTERESTED IN ONLY BITS 15:07
2453 014576 001411     TST R1 ;CHECK FOR ALL 0'S
2454 014600 013737 050516 050506     BEQ 9$ ;PASS
2455 014606 006237 050506     MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
2456 014612 104413     ASR CA121 ;PREPARE CA121 FOR TYPFOUT
                                ;ERROR
                                ;-----
014614 014612          .WORD -2
2457          ;TAG STORE ADDRESS LINE TESTS
    
```

```

2458
2459
2460 014616 050506          CA121
2461
2462
2463
2464 014620 000000
2465 014622 006337 050516 9$:  .WORD 0
2466 014626 022737 020000 050516 ASL  FLTPAT
2467 014634 001314          CMP  #20000,FLTPAT
2468 014636 000240          BNE  2$
      014640 005237 001472 10$:  NOP
      INC  $TESTN
    
```

```

:READING (HR<15:07> FOR CACHE TAG STORE
:DID NOT RESULT IN ALL 0'S
:PRINT CACHE TAG STORE ADDRESS LOCATION
:USED: CA<12:1>. NOTE THAT THE 1 IN
:THIS PATTERN WILL POINT TO THE ADDRESS
:LINE THAT POSSIBLY CAUSES ERROR.

:NEXT PATTERN
:HAS CACHE TAG STORE LOCAT. 4000 BEEN DONE?
:NO
:END OF TEST
:INCREMENT TEST COUNTER
    
```

2473

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

014644 000004

014646 014656
 014650 070004
 014652 000000
 014654 070042
 014656 012737
 014664 004437
 014670 014772

001015 177746
 002452

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

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

2474 014672 005037 060000
 2475 014676 112737 000002
 2476
 2477
 2478 014704 012737 000015
 2479 014712 005737 040000
 2480 014716 005737 060000
 2481
 2482
 2483 014722 005737 060000
 2484
 2485 014726 013737 177754
 2486 014734 000240
 014736 000240
 2487 014740 105037 177750
 2488 014744 012737 001015
 2489 014752 005737 050502
 2490 014756 001404
 2491 014760 104413

177750 1\$:
 177746
 177746
 050502 25\$:
 177746
 10\$:

```
        CLR 60000            ;0'S TO MAIN MEMORY LOCATION
        MOVB #HODO,CMR       ;ALLOWS CACHE DATA STORE BITS TO BE
                               ;WRITTEN TO CDR<15:0> ONLY DURING THE
                               ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
        MOV #15,CCR          ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
        TST 40000            ;
        TST 60000            ;WRITE ALL 0'S TO DATA STORE
                               ;LOCATION 0000 FROM MAIN MEMORY
        TST 60000            ;LOC. 60000
                               ;WRITE DATA STORE BITS FROM
        MOV CDR,CDR150       ;LOC. 0000 INTO CDR<15:0>.
        NOP                  ;SAVE CDR CONTENTS
        NOP                  ;INSTRUCTION 'JMP 1$' PLACED HERE
        CLRB CMR             ;FOR LOOP ON ERROR
        MOV #OFF,CCR         ;DISABLE MAINTENANCE
        TST CDR150           ;DISABLE CACHE
        BEQ 10$              ;CHECK FOR 0
        ERROR 10$           ;PASS
                               ;ERROR
        .WORD -2             ;-----
```

2492
 2493
 2494
 2495 014764 050502
 2496 014766 000000
 2497 014770 000240
 014772 005237 001472

```
        .WORD -2            ; DATA STORE TESTS
                               ;READING CDR<15:0> DID NOT RESULT
                               ;IN ALL 0'S
                               ;PRINT CDR<15:0> DATA READ.
        CDR150              ;
        .WORD 0              ;
        NOP                  ;END OF TEST
        INC $TESTN          ;INCREMENT TEST COUNTER
```

2502

.SBTTL TEST # 106 - WRITE ALL 1'S INTO DATA STORE LOCATION 0000

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

014776
014776 000004

015000 015010
015002 070006
015004 000000
015006 070044
015010 012737 001015 177746 40\$:
015016 004437 002452
015022 015130

TST106:
SCPCND :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
.WORD 40\$:TEST START LOCATION
.WORD 1\$-40\$+67764 :LOOP ON ERROR START LOCATION
.WORD 0 :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 25\$-40\$+67764 :LOOP ON ERROR END LOCATION
MOV #OFF,CCR :DISABLE CACHE
JSR R4,RELCTH :LOCATE TEST CODE TO HIGH CACHE SPACE
.WORD 10\$+2 :ADDRESS OF START OF NEXT TEST

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

2503 015024 012737 177777 060000
2504 015032 112737 000002 177750 1\$:
2505
2506
2507 015040 012737 000015 177746
2508 015046 005737 040000
2509 015052 005737 060000
2510
2511
2512 015056 005737 060000
2513
2514 015062 013737 177754 050502
2515 015070 000240
015072 000240 25\$:
2516 015074 105037 177750
2517 015100 012737 001015 177746
2518 015106 022737 177777 050502
2519 015114 001404
2520 015116 104413

015120 015116
2521
2522
2523
2524 015122 050502
2525 015124 000000
2526 015126 000240 10\$:
015130 005237 001472

MOV #-1,60000 :1'S TO MAIN MEMORY LOCATION
MOV# #HODO,CMR :ALLOWS CACHE DATA STORE BITS TO BE
:WRITTEN TO CDR<15:07> ONLY DURING THE
:DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
MOV #15,CCR :NO UCB SO AS TO WRITE ENABLE CACHE STORE
TST 40000
TST 60000 :WRITE ALL 1'S TO DATA STORE
:LOCATION 0000 FROM MAIN MEMORY
:LOC. 60000
TST 60000 :WRITE DATA STORE BITS FROM
:LOC. 0000 INTO CDR<15:07>.
MOV CDR,CDR150 :SAVE CDR CONTENTS
NOP :INSTRUCTION 'JMP 1\$' PLACED HERE
NOP :FOR LOOP ON ERROR
CLRB CMR :DISABLE MAINTENANCE
MOV #OFF,CCR :DISABLE CACHE
CMP #-1,CDR150 :CHECK ALL 1'S
BEQ 10\$:PASS
ERROR :ERROR
:-----
:DATA STORE TESTS
:READING CDR<15:0> DID NOT RESULT
:IN ALL 1'S
:PRINT CDR<15:0> DATA READ.
:END OF TEST
:INCREMENT TEST COUNTER

2531

```

.SBTTL TEST # 107 - WRITE FLOATING 1 PATRN INTO DATA STORE LOC 0
*****
*TEST 107      WRITE FLOATING 1 PATRN INTO DATA STORE LOC 0
*      WRITE FLOATING 1 PATTERN INTO DATA STORE LOCATION 0000.
*      READ FLOATING 1 PATTERN FROM CACHE DATA REGISTER.
*****
TST107:

```

```

015134
015134 000004          SCPCND          :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                :ERROR/LOOP ON TEST
015136 015146          .WORD 40$          :TEST START LOCATION
015140 070014          .WORD 1$-40$+67764 :LOOP ON ERROR START LOCATION
015142 000000          .WORD 0           :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
015144 070052          .WORD 25$-40$+67764 :LOOP ON ERROR END LOCATION
015146 012737 001015 177746 40$: MOV #OFF,CCR :DISABLE CACHE
015154 004437 002452 JSR R4,RELCTH :LOCATE TEST CODE TO HIGH CACHE SPACE
015160 015314          .WORD 10$+2       :ADDRESS OF START OF NEXT TEST

```

```

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

```

```

2532 015162 012737 000001 050516          MOV #1,FLTPAT :1ST FLOATING 1 PATTERN: 000001
2533 015170 013737 050516 060000 2$: MOV FLTPAT,60000 :FLOATING PATTERN TO MAIN MEMORY
2534 015176 112737 000002 177750 1$: MOVVB #HODO,CMR :ALLOWS CACHE DATA STORE BITS TO BE
2535                                     :WRITTEN TO CDR<15:07> ONLY DURING THE
2536                                     :DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2537 015204 012737 000015 177746          MOV #15,CCR :NO UCB SO AS TO WRITE ENABLE CACHE STORE
2538 015212 005737 040000          TST 40000 :
2539 015216 005737 060000          TST 60000 :WRITE FLOATING 1 PATTERN TO DATA STORE
2540                                     :LOCATION 0000 FROM MAIN MEMORY
2541                                     :LOC. 60000
2542 015222 005737 060000          TST 60000 :WRITE DATA STORE BITS FROM
2543                                     :LOC. 0000 INTO CDR<15:0>.
2544 015226 013737 177754 050502          MOV CDR,CDR150 :SAVE CDR CONTENTS
2545 015234 000240 25$: NOP :INSTRUCTION 'JMP 1$' PLACED HERE
2546 015240 105037 177750          CLRB CMR :FOR LOOP ON ERROR
2547 015244 012737 001015 177746          MOV #OFF,CCR :DISABLE MAINTENANCE
2548 015252 023737 050516 050502          CMP FLTPAT,CDR150 :DISABLE CACHE
2549 015260 001410          BEQ 9$ :CHECK FOR CORRECT PATTERN
2550 015262 013737 050516 050504          MOV FLTPAT,EXDAT6 :PASS
2551 015270 104413          ERROR :SAVE FLOATING PATTERN FOR TYPEOUT
                                     :ERROR
                                     :-----
015272 015270          .WORD -2
2552                                     : DATA STORE TESTS
2553                                     :READING CDR<15:0> DID NOT RESULT
2554                                     :IN CORRECT FLOATING 1 PATTERN
2555 015274 050504          EXDAT6 :PRINT FLOATING PATTERN EXPECTED
2556 015276 050502          CDR150 :PRINT CDR<15:0> DATA READ.
2557 015300 000000          .WORD 0
2558 015302 006337 050516 9$: ASL FLTPAT :NEXT PATTERN
2559 015306 103401          BCS 10$ :IF PATTERN 100000 HAS BEEN DONE;FINISHED
2560 015310 000727          BR 2$ :IF NOT, NEXT PASS
2561 015312 000240 10$: NOP :END OF TEST
015314 005237 001472          INC $TESTN :INCREMENT TEST COUNTER

```

2566

```

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

```

```

015320
015320 G00004
:SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
015322 015332 .WORD 40$ ;TEST START LOCATION
015324 070024 .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
015326 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
015330 070054 .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
015332 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
015340 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
015344 015512 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

2567 015346 012705 060000 MOV #60000,R5 ;ADDRESS 60000 INTO R5
2568 015352 005025 5$: CLR (R5)+ ;CLEAR MAIN MEMORY LOW CACHE AREA
2569 015354 020527 070000 CMP R5,#70000 ;FINISHED?
2570 015360 001374 BNE 5$ ;NO
2571 015362 012705 060000 MOV #60000,R5 ;START WITH ADDRESS 60000
2572 015366 012703 040000 MOV #40000,R3 ;ADDRESS 40000 INTO R3
2573 015372 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE
2574 WRITTEN TO CDR<15:0> ONLY DURING THE
2575 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2576 015400 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2577 015406 005713 TST (R3) ;
2578 015410 005715 TST (R5) ;WRITE ALL 0'S TO DATA STORE FROM MAIN MEM.
2579 015412 005715 TST (R5) ;WRITE DATA STORE BITS INTO CDR<15:0>
2580 015414 013737 177754 050502 MOV CDR,CDR150 ;SAVE CDR CONTENTS
2581 015422 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
015424 000240 NOP ;FOR LOOP ON ERROR
2582 015426 105037 177750 CLRB CMR ;DISABLE MAINTENANCE
2583 015432 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
2584 015440 022737 000000 050502 CMP #0,CDR150 ;CHECK ALL 0'S
2585 015446 001411 BEQ 9$ ;PASS
2586 015450 010537 050466 MOV R5,CA210+2 ;SAVE ADDRESS USED THIS PASS
2587 015454 005037 050464 CLR CA210 ;
2588 015460 104413 ERROR ;ERROR
;-----
015462 015460 .WORD -2 ;
2589 ; DATA STORE TESTS
2590 ;READING CDR<15:0> DID NOT RESULT
2591 ;IN ALL 0'S
2592 015464 050464 CA210 ;PRINT CA<21:0> USED
2593 015466 050502 CDR150 ;PRINT CDR<15:0> DATA READ.
2594 015470 000000 .WORD 0 ;
2595 015472 062705 000002 9$: ADD #2,R5 ;NEXT CACHE LOCATION
2596 015476 062703 000002 ADD #2,R3 ;
2597 015502 022705 070000 CMP #70000,R5 ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
2598 015506 001331 BNE 1$ ;NO
2599 015510 000240 10$: NOP ;END OF TEST
015512 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```

2604

```

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

```

015516
015516 000004          SCLPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
015520 015530          .WORD 405          ;TEST START LOCATION
015522 070026          .WORD 1$-40$+67764      ;LOOP ON ERROR START LOCATION
015524 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
015526 070056          .WORD 25$-40$+67764    ;LOOP ON ERROR END LOCATION
015530 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
015536 004437 002452     JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
015542 015712          .WORD 10$+2          ;ADDRESS OF START OF NEXT TEST

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

2605 015544 012705 060000          MOV #60000,R5 ;ADDRESS 60000 INTO R5
2606 015550 012725 177777          5$: MOV #-1,(R5)+ ;1'S TO MAIN MEMORY LOW CACHE AREA
2607 015554 020527 070000          CMP R5,#70000 ;FINISHED?
2608 015560 001373          BNE 5$ ;NO
2609 015562 012705 060000          MOV #60000,R5 ;START WITH ADDRESS 60000
2610 015566 012703 040000          MOV #40000,R3 ;ADDRESS 40000 INTO R3
2611 015572 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE
2612                                     ;WRITTEN TO CDR<15:0> ONLY DURING THE
2613                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2614 015600 012737 000015 177746          MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2615 015606 005713          TST (R3) ;
2616 015610 005715          TST (R5) ;WRITE ALL 1'S TO DATA STORE FROM MAIN MEM.
2617 015612 005715          TST (R5) ;WRITE DATA STORE BITS INTO CDR<15:0>
2618 015614 013737 177754 050502          MOV CDR,CDR150 ;SAVE CDR CONTENTS
2619 015622 000240          25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
2620 015624 000240          NOP ;FOR LOOP ON ERROR
2621 015626 105037 177750          CLRB CMR ;DISABLE MAINTENANCE
2622 015632 012737 001015 177746          MOV #OFF,CCR ;DISABLE CACHE
2623 015640 022737 177777 050502          CMP #-1,CDR150 ;CHECK ALL 1'S
2624 015646 001411          BEQ 9$ ;PASS
2625 015650 010537 050466          MOV R5,CA210+2 ;SAVE ADDRESS USED THIS PASS
2626 015654 005037 050464          CLR CA210 ;
2627 015662 015660          .WORD -2 ;ERROR
2628                                     ;-----
2629                                     ; DATA STORE TESTS
2630 015664 050464          CA210 ;READING CDR<15:0> DID NOT RESULT
2631 015666 050502          CDR150 ;IN ALL 1'S
2632 015670 000000          .WORD 0 ;PRINT CA<21:0> USED
2633 015672 062705 000002          9$: ADD #2,R5 ;PRINT CDR<15:0> DATA READ.
2634 015676 062703 000002          ADD #2,R3 ;NEXT CACHE LOCATION
2635 015702 022705 070000          CMP #70000,R5 ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
2636 015706 001331          BNE 1$ ;NO
2637 015710 000240          10$: NOP ;END OF TEST
2637 015712 005237 001472          INC $TESTN ;INCREMENT TEST COUNTER
    
```



```

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

```

015716
015716 000004          SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
015720 015730          .WORD 40$          ;TEST START LOCATION
015722 060024          .WORD 1$-40$+57764        ;LOOP ON ERROR START LOCATION
015724 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
015726 060054          .WORD 25$-40$+57764    ;LOOP ON ERROR END LOCATION
015730 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
015736 004437 002424   JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
015742 016110          .WORD 10$+2        ;ADDRESS OF START OF NEXT TEST
    
```

```

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

```

2643 015744 012705 070000          MOV #70000,R5 ;ADDRESS 70000 INTO R5
2644 015750 005025          CLR (R5)+      ;CLEAR MAIN MEMORY HI CACHE AREA
2645 015752 020527 100000          CMP R5,#100000 ;FINISHED?
2646 015756 001374          BNE 5$        ;NO
2647 015760 012705 070000          MOV #70000,R5 ;START WITH ADDRESS 70000
2648 015764 012703 050000          MOV #50000,R3 ;ADDRESS 50000 INTO R3
2649 015770 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE
2650                                     ;WRITTEN TO CDR<15:0> ONLY DURING THE
2651                                     ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2652 015776 012737 000015 177746   MOV #15,CCR    ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2653 016004 005713          TST (R3)      ;
2654 016006 005715          TST (R5)      ;WRITE ALL 0'S TO DATA STORE FROM MAIN MEM.
2655 016010 005715          TST (R5)      ;WRITE DATA STORE BITS INTO CDR<15:0>
2656 016012 013737 177754 050502   MOV CDR,CDR150 ;SAVE CDR CONTENTS
2657 016020 000240          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
2658 016022 000240          NOP          ;FOR LOOP ON ERROR
2658 016024 105037 177750          CLR# CMR     ;DISABLE MAINTENANCE
2659 016030 012737 001015 177746   MOV #OFF,CCR ;DISABLE CACHE
2660 016036 022737 000000 050502   CMP #0,CDR150 ;CHECK ALL 0'S
2661 016044 001411          BEQ 9$        ;PASS
2662 016046 010537 050466          MOV R5,CA210+2 ;SAVE ADDRESS USED THIS PASS
2663 016052 005037 050464          CLR CA210    ;
2664 016056 104413          ERROR          ;ERROR
2665                                     ;-----
2665 016060 016056          .WORD -2        ; DATA STORE TESTS
2666                                     ;READING CDR<15:0> DID NOT RESULT
2667                                     ;IN ALL 0'S
2668 016062 050464          CA210        ;PRINT CA<21:0> USED
2669 016064 050502          CDR150       ;PRINT CDR<15:0> DATA READ.
2670 016066 000000          .WORD 0
2671 016070 062705 000002          9$: ADD #2,R5    ;NEXT CACHE LOCATION
2672 016074 062703 000002          ADD #2,R3
2673 016100 022705 100000          CMP #100000,R5 ;HAVE ALL HIGH CACHE LOCATIONS BEEN DONE?
2674 016104 001331          BNE 1$        ;NO
2675 016106 000240          10$: NOP      ;END OF TEST
2675 016110 005237 001472          INC $TESTN   ;INCREMENT TEST COUNTER
    
```

2680

```

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

```

```

016114
016114 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
016116 016126          .WORD 40$          ;TEST START LOCATION
016120 060026          .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
016122 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
016124 060056          .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
016126 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
016134 004437 002424 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
016140 016310          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

2681 016142 012705 070000          MOV #70000,R5 ;ADDRESS 70000 INTO R5
2682 016146 012725 177777          MOV #-1,(R5)+ ;1'S TO MAIN MEMORY HI CACHE AREA
2683 016152 020527 100000          CMP R5,#100000 ;FINISHED?
2684 016156 001373          BNE 5$ ;NO
2685 016160 012705 070000          MOV #70000,R5 ;START WITH ADDRESS 70000
2686 016164 012703 050000          MOV #50000,R3 ;ADDRESS 50000 INTO R3
2687 016170 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS CACHE DATA STORE BITS TO BE
2688 ;WRITTEN TO CDR<15:0> ONLY DURING THE
2689 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2690 016176 012737 000015 177746          MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2691 016204 005713          TST (R3) ;
2692 016206 005715          TST (R5) ;WRITE ALL 1'S TO DATA STORE FROM MAIN MEM.
2693 016210 005715          TST (R5) ;WRITE DATA STORE BITS INTO CDR<15:0>
2694 016212 013737 177754 050502          MOV CDR,CDR150 ;SAVE CDR CONTENTS
2695 016220 000240          NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
2696 016222 000240          NOP ;FOR LOOP ON ERROR
2696 016224 105037 177750          CLRB CMR ;DISABLE MAINTENANCE
2697 016230 012737 001015 177746          MOV #OFF,CCR ;DISABLE CACHE
2698 016236 022737 177777 050502          CMP #-1,CDR150 ;CHECK ALL 1'S
2699 016244 001411          BEQ 9$ ;PASS
2700 016246 010537 050466          MOV R5,CA210+2 ;SAVE ADDRESS USED THIS PASS
2701 016252 005037 050464          CLR CA210 ;
2702 016256 104413          ERROR ;ERROR
                                ;-----
016260 016256          .WORD -2 ;
2703 ; DATA STORE TESTS
2704 ;READING CDR<15:0> DID NOT RESULT
2705 ;IN ALL 1'S
2706 016262 050464          CA210 ;PRINT CA<21:0> USED
2707 016264 050502          CDR150 ;PRINT CDR<15:0> DATA READ.
2708 016266 000000          .WORD 0 ;
2709 016270 062705 000002          9$: ADD #2,R5 ;NEXT CACHE LOCATION
2710 016274 062703 000002          ADD #2,R3 ;
2711 016300 022705 100000          CMP #100000,R5 ;HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
2712 016304 001331          BNE 1$ ;NO
2713 016306 000240          10$: NOP ;END OF TEST
016310 005237 001472          INC $TESTN ;INCREMENT TEST COUNTER

```

2717

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

```
016314
016314 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
016316 016326          .WORD 40$          ;TEST START LOCATION
016320 070040          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
016322 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
016324 070100          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
016326 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
016334 004437 002452     JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
016340 016532          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST
```

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

```
2718 016342 000240          NOP
2719
2720
2721 016344 005037 060000          CLR 60000 ;WHEN THE TEST IS RELOCATED,ADDRESS 70000
2722 016350 012737 000002 050516 MOV #2,FLTPAT ;WILL BE LOADED WITH ALL 1'S DURING TEST
2723 016356 012702 040000          MOV #40000,R2 ;ALL 0'S TO MAIN MEMORY
2724 016362 012703 060000          MOV #60000,R3 ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
2725 016366 063702 050516          ADD FLTPAT,R2 ;
2726 016372 063703 050516          ADD FLTPAT,R3 ;R2 CONTAINS 40000+FLTPAT
2727 016376 012713 177777          MOV #-1,(R3) ;R3 CONTAINS 60000+FLTPAT
2728
2729 016402 112737 000002 177750 1$: MOVB #HODO,CMR ;ALL 1'S TO MAIN MEM. LOCATION
2730
2731
2732 016410 012737 000015 177746 MOV #15,CCR ;SPECIFIED BY R3
2733 016416 005737 040000          TST 40000 ;ALLOWS CACHE DATA STORE BITS TO BE
2734 016422 005737 060000          TST 60000 ;WRITTEN TO CDR<15:0> ONLY DURING THE
2735
2736 016426 005712          TST (R2) ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
2737 016430 005713          TST (R3) ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2738
2739
2740 016432 005737 060000          TST 60000 ;WRITE ALL 0'S FROM MAIN MEM. LOCATION
2741
2742 016436 013701 177754          MOV CDR,R1 ;60000 INTO CACHE DATA STORE LOCATION 0000.
2743 016442 000240          NOP ;LOAD DATA FROM CACHE DATA STORE LOCATION
016444 000240          NOP ;0000 INTO CDR.
2744 016446 105037 177750          CLR# CMR ;SAVE CDR DATA
2745 016452 012737 001015 177746 MOV #OFF,CCR ;INSTRUCTION 'JMP 1$' PLACED HERE
2746 016460 005701          TST R1 ;FOR LOOP ON ERROR
2747 016462 001411          BEQ 9$ ;DISABLE MAINT. MODE
2748
2749 016464 013737 050516 050506 MOV FLTPAT,CA121 ;DISABLE CACHE
2750 016472 006237 050506 ASR CA121 ;CHECK FOR ALL 0'S
2751 016476 104413          ERROR ;PASS
;-----
016500 016476          .WORD -2 ;SAVE CA<12:1> USED
;PREPARE CA121 FOR TYPEOUT
;ERROR
;-----
;DATA STORE TESTS- ADDRESS LINE VERIFICATION
```


2772

```

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

```

016536
016536 000004

016540 016550
016542 070000
016544 000000
016546 070036
016550 012737 001015 177746 40$:
016556 004437 002452
016562 016726
  
```

```

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

```

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

```

2773 016564 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS COMPARED RESULTS TO BE
2774 ;WRITTEN TO CMR<15:13> ONLY DURING
2775 ;THE DESTINATION MEMORY ACCESS
2776 ;OF AN INSTRUCTION
2777 016572 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
2778 016600 005737 040000 TST 40000
2779 016604 005737 000000 TST 0 ;WRITE ALL 0'S INTO TAG STORE LOCATION 0000
2780 ;FROM CACHE ADDRESS CA<21:13>
2781 016610 005737 000000 TST 0 ;PLACE ALL 0'S ON CA<21:13> FOR COMPARISON
2782 ;WITH TAG STORE DATA. WRITE COMPARED
2783 ;RESULTS INDICATION IN CMR<15:13>
2784 016614 013737 177750 050512 MOV CMR,CM1513 ;SAVE CMR DATA
2785 016622 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
016624 000240 NOP ;FOR LOOP ON ERROR
2786 016626 105037 177750 CLRB CMR ;DISABLE MAINTENANCE MODE
2787 016632 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
2788 016640 042737 017777 050512 BIC #17777,CM1513
2789 016646 022737 160000 050512 CMP #160000,CM1513 ;CHECK THAT CMR<15:13> ALL 1'S
2790 016654 001423 BEQ 10$ ;PASS
2791 016656 012737 000007 050510 MOV #7,EXDAT1 ;INDICATE EXPECTED CMR<15:13>
2792 016664 012737 000015 002062 MOV #13,LOOP ;ERROR;PREPARE CM1513. FOR TYPEOUT
2793 016672 006237 050512 2$: ASR CM1513
2794 016676 042737 100000 050512 BIC #100000,CM1513
2795 016704 005337 002062 DEC LOOP
2796 016710 001370 BNE 2$
2797 016712 104413 ERROR ;ERROR
;-----
016714 016712 .WORD -2
2798 ;COMPARE TAG STORE & CA<21:13> TESTS
2799 ;BITS 15 THRU 13 OF CMR DID NOT READ
2800 ;AS ALL 1'S
2801 016716 050510 EXDAT1 ;PRINT CMR<15:13> DATA EXPECTED
2802 016720 050512 CM1513 ;PRINT CMR<15:13> DATA RECEIVED
2803 016722 000000 .WORD 0
  
```

2804	016724	000240	08:	NOP	:	END OF TEST
	016726	005237		INC	:	INCREMENT TEST COUNTER

2812

.SBTTL TEST # 116 - UNEQUAL DATA COMPARISON CONDITION CAN BE DETECTED

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

016732
 016732 000004

 016734 016744
 016736 070036
 016740 000000
 016742 070130
 016744 012737 001015 177746
 016752 004437 002452
 016756 017320

TST116:
 SPCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 ;ERROR/LOOP ON TEST
 .WORD 40\$;TEST START LOCATION
 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION
 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION
 40\$: MOV #OFF,CCR ;DISABLE CACHE
 JSR R4,RELCBH ;LOCATE TEST CODE TO HIGH CACHE SPACE
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

2813 016760 000411
 2814
 2815
 2816 016762 000006
 2817 016764 000005
 2818 016766 000005
 2819 016770 000005
 2820 016772 000005
 282 016774 000003
 2822 016776 000003
 2823 017000 000003
 2824 017002 000003
 2825 017004 012737 000200 172350
 2826
 2827
 2828
 2829
 2830
 2831 017012 012701 016762
 2832
 2833 017016 012737 070300 000004
 2834 017024 012737 000340 000006
 2835 017032 112737 000002 177750
 2836
 2837
 2838 017040 012737 000015 177746
 2839 017046 012737 000001 177572
 2840 017054 012737 000020 172516
 2841 017062 005737 040000
 2842 017066 005737 000000
 2843 017072 005737 100000
 2844
 2845
 2846

2\$: BR 39\$;BRANCH OVER TABLE
 CMR<15:13> CA<21:13> TAG STORE

 30\$: .WORD 6 ; 001 000
 31\$: .WORD 5 ; 002 000
 32\$: .WORD 5 ; 004 000
 33\$: .WORD 5 ; 010 000
 34\$: .WORD 5 ; 020 000
 35\$: .WORD 3 ; 040 000
 36\$: .WORD 3 ; 100 000
 37\$: .WORD 3 ; 200 000
 38\$: .WORD 3 ; 400 000
 39\$: MOV #200,KPAR4 ;KPAR4 CONTAINS THE FLOATING 1 PATTERN
 ;AND REPRESENTS THE THE PAGE ADDRESS FIELD
 ;DATA USED BY MEMORY MNGMNT. TO CONSTRUCT
 ;THE PHYSICAL ADDRESS. 200 IS THE 1ST
 ;FLOATING 1 PATTERN WHICH WILL BE CONSTRUCTED
 ;TO ADDRESS 20000.
 ;SAVE ADDRESS OF FIRST CMR<15:13>
 ;EXPECTED DATA
 1\$: MOV #7\$-2\$+70000,4 ;ALLOW FOR NEX TRAP
 MOV #340,6
 MOV #HODO,CMR ;ALLOWS COMPARED RESULTS TO BE WRITTEN
 ;TO CMR<15:13> ONLY DURING THE DESTINATION
 ;MEMORY ACCESS OF AN INSTRUCTION.
 ;NO UCB SO AS TO WRITE CACHE STORE
 MOV #1,SRO ;ENABLE MEMORY MNGMNT.
 MOV #20,SR3 ;ENABLE 22-BIT MAPPING
 TST 40000
 TST 0 ;WRITE ALL 0'S TO TAG STORE LOCATION 0000
 TST 100000 ;CHOOSES KPAR4 FOR ADDRESSING. PHYSICAL
 ;ADDRESS WILL BE DETERMINED BY FLOATING
 ;PATTERN USED IN KPAR4.
 ;PLACE DLOATING PATTERN ON CA<21:13>

```

2847
2848
2849
2850 017076 000240      NOP
2851 017100 000240      NOP
2852 017102 013737 177750 050512      MOV      CMR,CM1513      ;FOR COMPARISON WITH TAG STORE DATA
2853 017110 000240      NOP      ;AT LOCATION 0000. WRITE THE COMPARED
017112 000240      NOP      ;RESULT INDICATION INTO CMR<15:13>.
2854 017114 005037 177572      CLR      SR0      ;SAVE CMR DATA
2855 017120 005037 172516      CLR      SR3      ;INSTRUCTION 'JMP 1$' PLACED HERE
2856 017124 012737 001015 177746      MOV      #OFF,CCR      ;FOR LOOP ON ERROR
2857 017132 105037 177750      CLRB    CMR      ;DISABLE MEM MNGMENT.
2858 017136 012737 000015 002062      MOV      #13,LOOP      ;DISABLE CACHE
2859 017144 006237 050512 5$:      ASR      CM1513      ;DISABLE MAINTENANCE
2860 017150 042737 100000 050512      BIC      #100000,CM1513 ;PREPARE CM1513. FOR ERROR CHECK
2861 017156 005337 002062      DEC      LOOP
2862 017162 001370      BNE     5$
2863 017164 023711 050512      CMP      CM1513,(R1)    ;COMPARE CMR<15:13> RECEIVED WITH
2864                                     ;EXPECTED
2865 017170 001426      BEQ     9$      ;PASS
2866 017172 013737 172350 050476      MOV      KPAR4,CA2113 ;SAVE PATTERN USED FOR CA<21:13>
2867 017200 012737 000007 002062      MOV      #7,LOOP      ;PREPARE CA2113 FOR PRINTOUT
2868 017206 006237 050476 6$:      ASR      CA2113
2869 017212 042737 000000 050476      BIC      #100000,CA2113
2870 017220 005337 002062      DEC      LOOP
2871 017224 001370      BNE     6$
2872 017226 011137 050510      MOV      (R1),EXDAT1   ;PREPARE EXPECTED FOR PRINTOUT
2873 017232 104413      ERROR   ;ERROR
017234 017232      .WORD  -2      ;-----
2874                                     ;COMPARE TAG STORE AND CA<21:13> TESTS
2875                                     ;CMR<15:13> DID NOT READ CORRECTLY
2876 017236 050476      CA2113 ;PRINT CA<21:13> PATTERN USED ON ADDRESS
2877                                     ;LINES
2878 017240 050510      EXDAT1 ;PRINT CMR<15:13> EXPECTED
2879 017242 050512      CM1513 ;PRINT CMR<15:13> RECEIVED
2880 017244 000000      .WORD  0      ;PRINT TERMINATE
2881 017246 006337 172350 9$:      ASL      KPAR4      ;NEXT PATTERN; IF PHYSICAL ADDRESS
2882                                     ;10000000 HAS BEEN DONE; FINISHED
2883 017252 103414      BCS     8$
2884 017254 005721      TST     (R1)+      ;
2885                                     ;NOT FINISHED; POINT TO NEXT
2886 017256 000657      BR      1$      ;CMR<15:13> EXPECTED
2887 017260 005037 177572 7$:      CLR      SR0      ;CONTINUE WITH TEST
2888 017264 005037 172516      CLR      SR3      ;DISABLE MEM. MNGMENT.
2889 017270 105037 177750      CLRB    CMR      ;DISABLE MAINTENANCE
2890 017274 012737 001015 177746      MOV      #OFF,CCR      ;DISABLE CACHE
2891 017302 022626      CMP     (SP)+,(SP)+ ;RESTORE STACK DUE TO INTERRUPT
2892 017304 012737 000006 000004 8$:      MOV      #6,4      ;RESTORE VECTORS
2893 017312 005037 000006      CLR     6
2894 017316 000240 10$:      NOP
017320 005237 001472      INC     $TESTN      ;END OF TEST
;INCREMENT TEST COUNTER

```


..SBTTL TEST # 117 - TEST FLUSH IN PROGRESS BIT

..TEST 117 TEST FLUSH IN PROGRESS BIT
..VERIFY FLUSH IN PROGRESS BIT WILL SET AS A RESULT OF FLUSH

..TEST117:

017324	017324	000004				SCPCND		:SCOPE CONDITIONS:GO SET UP FOR LOOP ON	
								:ERROR/LOOP ON TEST	
								:TEST START LOCATION	
017326	017336					.WORD	40\$:LOOP ON ERROR START LOCATION	
017330	017336					.WORD	1\$:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST	
017332	000000					.WORD	0	:LOOP ON ERROR END LOCATION	
017334	017372					.WORD	25\$		
017336									
2899	017336	005002				40\$:			
						1\$:	CLR	R2	:INITIALIZE COUNTER
2900	017340	005037	00207C				CLR	FAIL1	:INITIALIZE ERROR FLAG
2901	017344	052737	000400	177746			BIS	#FC,CCR	:FLUSH CACHE
2902	017352	032737	010000	177746			BIT	#VCIP,CCR	:VERIFY FLUSH IN PROGRESS
2903	017360	001002					BNE	3\$:VCIP BIT IS SET
2904	017362	005237	002070				INC	FAIL1	:INDICATE ERROR
2905	017366	005302				3\$:	DEC	R2	:WAIT DELAY FOR FLUSH TO COMPLETE
2906	017370	001376					BNE	3\$	
2907	017372	000240				25\$:	NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	017374	000240					NOP		:FOR LOOP ON ERROR
2908	017376	005737	002070				TST	FAIL1	:IS THERE ERROR
2909	017402	001403					BEQ	10\$:PASS
2910	017404	104413					ERROR		:ERROR
									:-----
	017406	017404					.WORD	-2	
2911									:FLUSH CACHE TESTS
2912									:FLUSH IN PROGRESS BIT(VCIP) FAILED
2913									:TO SET AS A RESULT OF SETTING CACHE FLUSH BIT
2914	017410	000000					.WORD	0	
2915	017412	000240				10\$:	NOP		:END OF TEST
	017414	005237	001472				INC	\$TESTN	:INCREMENT TEST COUNTER

TEST # 120 - TEST FLUSH IN PROGRESS BIT(VCIP) WILL RESET

2919

```

.SBTTL TEST # 120 - TEST FLUSH IN PROGRESS BIT(VCIP) WILL RESET
:.....
:TEST 120 TEST FLUSH IN PROGRESS BIT(VCIP) WILL RESET
:* VERIFY FLUSH IN PROGRESS BIT(VCIP) WILL RESET ON COMPLETION OF FLUSH
:.....
TST120:
      SCPCMD           ;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
017420 000004
017422 017432          .WORD 40$
017424 017432          .WORD 1$
017426 000000          .WORD 0
017430 017456          .WORD 25$
017432          40$:
2920 017432 005002      1$: CLR R2           ;INITIALIZE COUNTER
2921 017434 052737 000400 177746 1$: BIS #FC,CCR      ;START FLUSH
2922 017442 032737 010000 177746 3$: BIT #VCIP,CCR    ;SEE IF FLUSH COMPLETE
2923 017450 001407      BEQ 10$           ;FLUSH COMPLETE
2924 017452 005302      DEC R2           ;SEE IF TIME HAS RUN OUT
2925 017454 001372      BNE 3$           ;NOT YET
2926 017456 000240      25$: NOP             ;INSTRUCTION 'JMP 1$' PLACED HERE
2927 017462 104413      NOP             ;FOR LOOP ON ERROR
                        ERROR
                        ;ERROR
                        ;-----
017464 017462          .WORD -2
2928
2929
2930
2931 017466 000000          .WORD 0
2932 017470 000240          10$: NOP
2933 017472 005237 001472      INC $TESTN      ;END OF TEST
                        ;INCREMENT TEST COUNTER

```

2928
2929
2930

2937

```

.SBTTL TEST # 121 - CHECK THAT VSIU BIT SETS
.....
*TEST 121 CHECK THAT VSIU BIT SETS
* VERIFY THAT VSIU BIT WILL CHANGE FROM A CLEAR TO SET CONDITION AS
* A RESULT OF CACHE FLUSH
.....
TST121:
      SCPCND                :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                              :ERROR/LOOP ON TEST
                              :TEST START LOCATION
      .WORD 40$              :LOOP ON ERROR START LOCATION
      .WORD 1$               :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0                :LOOP ON ERROR END LOCATION
      .WORD 25$

2938 017510 032737 020000 177746 40$: BIT #VSIU,CCR :IS SET A BEING USED
2939 017516 001407          BEQ 3$ :YES
2940 017520 052737 000400 177746 BIS #FC,CCR :CAUSE FLUSH FOR SET A
2941 017526 032737 010000 177746 200$: BIT #VCIP,CCR :WAIT FOR FLUSH TO COMPLETE
2942 017534 001374          BNE 200$
2943 017536 052737 000400 177746 3$: BIS #FC,CCR :CAUSE FLUSH
2944 017544 032737 010000 177746 4$: BIT #VCIP,CCR :WAIT FOR FLUSH TO COMPLETE
2945 017552 001374          BNE 4$
2946 017554 000240          NOP :INSTRUCTION 'JMP 1$' PLACED HERE
      017556 000240          NOP :FOR LOOP ON ERROR
2947 017560 032737 020000 177746 BIT #VSIU,CCR :IS VSIU BIT -1 INDICATING VALID SET
2948          :B WAS SELECTED
2949 017566 001003          BNE 10$ :PASS
2950 017570 104413          ERROR :ERROR
      :-----
      017572 017570          .WORD -2
2951          :FLUSH CACHE TESTS
2952          :VSIU BIT DID NOT SET AS A RESULT OF FLUSH
2953 017574 000000          .WORD 0
2954 017576 000240          NOP :END OF TEST
      017600 005237 001472 INC $TESTN :INCREMENT TEST COUNTER
  
```

2959

```

.SBTTL TEST # 122 - CHECK THAT VSIU BIT CLEARS
:.....
:TEST 122 CHECK THAT VSIU BIT CLEARS
:  VERIFY THAT VSIU BIT WILL CHANGE FROM A SET TO CLEAR CONDITION AS
:  A RESULT OF CACHE FLUSH
:.....
YST122:
      SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                        ;ERROR/LOOP ON TEST
                        ;TEST START LOCATION
      .WORD 40$       ;LOOP ON ERROR START LOCATION
      .WORD 1$        ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0         ;LOOP ON ERROR END LOCATION
      .WORD 25$
2960 017616 032737 020000 177746 1$: BIT #VSIU,CCR ;IS SET B BEING USED
2961 017624 001007          BNE 3$ ;YES
2962 017626 052737 000400 177746 BIS #FC,CCR ;CAUSE FLUSH FOR SET B
2963 017634 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT FOR FLUSH TO COMPLETE
2964 017642 001374          BNE 200$
2965 017644 052737 000400 177746 3$: BIS #FC,CCR ;CAUSE FLUSH
2966 017652 032737 010000 177746 4$: BIT #VCIP,CCR ;WAIT FOR FLUSH TO COMPLETE
2967 017660 001374          BNE 4$
2968 017662 000240          NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
      017664 000240          NOP ;FOR LOOP ON ERROR
2969 017666 032737 020000 177746 BIT #VSIU,CCR ;IS VSIU BIT -0 INDICATING VALID SET
2970          ;A WAS SELECTED
2971 017674 001403          BEQ 10$ ;PASS
2972 017676 104413          ERROR ;ERROR
      ;-----
      .WORD -2          ;FLUSH CACHE TESTS
                        ;VSIU BIT DID NOT CLEAR AS A RESULT OF FLUSH
2973          ;
2974          ;
2975 017702 000000          .WORD 0
2976 017704 000240          10$: NOP ;END OF TEST
      017706 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
  
```

2982

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

017712
 017712 000004
 017714 017724
 017716 070052
 017720 000000
 017722 070100
 017724 012737 001015 177746 40\$:
 017732 004437 002452
 017736 020124

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

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

2983 017740 032737 020000 177746
 2984 017746 001407
 2985 017750 052737 000400 177746
 2986 017756 032737 010000 177746 200\$:
 2987 017764 001374
 2988 017766 012737 177777 177752 3\$:
 2989
 2990 017774 112737 000374 177751
 2991 020002 012705 060000
 2992 020006 012703 040000
 2993 020012 012737 000015 177746 1\$:
 2994 020020 112737 000003 177750
 2995
 2996
 2997
 2998
 2999
 3000 020026 005713
 3001 020030 005715
 3002
 3003 020032 005715
 3004
 3005
 3006 020034 013701 177750
 3007 020040 000240 25\$:
 020042 000240
 3008 020044 105037 177750
 3009 020050 012737 001015 177746
 3010 020056 032701 010000
 3011 020062 001410
 3012 020064 010537 050506
 3013
 3014 020070 006237 050506
 3015 020074 104413
 020076 020074

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

TEST # 123 - WRITE AND READ 0'S TO ALL LOW CACHE VALID

```

3016
3017
3018
3019 020100 050506
3020
3021 020102 000000
3022 020104 062705 000002
3023 020110 062703 000002
3024 020114 020527 070000
3025 020120 001334
3026 020122 000240
      020124 005237 001472
    
```

```

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

```

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

:NEXT VALID STORE LOCATION

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

3 52

```

.SBTTL TEST # 124 - WRITE AND READ 1'S TO ALL LOW CACHE VALID
*****
TEST 124 WRITE AND READ 1'S TO ALL LOW CACHE VALID
WRITE AND READ 1'S TO ALL LOW CACHE VALID
BIT STORE ADDRESS LOCATIONS- SET A
(VVALID STORE LOCATIONS 0000 TO 3777)
*****
TEST124:
020130 000004 SPCOND ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
020130 000004 ;ERROR/LOOP ON TEST
020132 020142 .WORD 40$ ;TEST START LOCATION
020134 020206 .WORD 1$ ;LOOP ON ERROR START LOCATION
020136 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
020140 020234 .WORD 25$ ;LOOP ON ERROR END LOCATION
020142 40$:
3033 020142 012737 000124 007472 MOV #124,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
3034 020150 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
3035 020156 001407 BEQ 3$ ;YES
3036 020160 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A
3037 020166 032737 010000 177746 BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3038 020174 001374 BNE 200$
3039 020176 012705 060000 3$: MOV #60000,R5 ;;ADDRESS 60000 INTO R5
3040 020202 012703 040000 MOV #40000,R3 ;ADDRESS 40000 INTO R3
3041 020206 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3042 020214 112737 000002 177750 M' B #HODO,CMR ;HODO ALLOWS VALID STORE SET A TO
;BE WRITTEN TO CMR<12> ONLY DURING
;THE DESTINATION MEMORY ACCESS.
3043
3044 020222 005713 TST (R3)
3045 020224 005715 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3046 020226 005715 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
;FROM VALID STORE ADDRESS LOCATION
;JUST WRITTEN INTO.
3047
3048
3049
3050 020230 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
3051 020234 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
020236 000240 NOP ;FOR LOOP ON ERRCR
3052 020240 105037 177750 CLR# CMR ;DISABLE MAINT. MODE
3053 020244 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3054 020252 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.
3055 020256 001010 BNE 9$ ;PASS
3056
3057 020260 010537 050506 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
3058 ;USED: CA<12:1>
3059 020264 006237 050506 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3060 020270 104413 ERROR ;ERROR
;-----
020272 020270 .WORD -2
3061 ;VALID BITS STORE TESTS
3062 ;READING VALID STORE DATA SET A
3063 ;THRU CMR<12> DID NOT RESULT IN 1.
3064 020274 050506 CA121 ;PRINT VALID STORE ADDRESS LOCATION
3065 ;USED: CA<12:1>.
3066 020276 000000 .WORD 0
3067 020300 062705 000002 9$: ADD #2,R5 ;NEXT VALID STORE LOCATION
3068 020304 062703 000002 ADD #2,R3
3069 020310 020527 070000 CMP R5,#70000 ;HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
3070 020314 001334 BNE 1$ ;NO

```

TRKACO 11-44 KK11B CACHE MACRO M1113 28-MAR-81 14:20 PAGE 106-1 L 10
TEST # 124 - WRITE AND READ 1'S TO ALL LOW CACHE VALID

SEQUENCE 128

3.71 020316 000240
020320 005237 C01472

108: NOP
INC \$TESTN

:END OF TEST
:INCREMENT TEST COUNTER

3077

```
.SBTTL TEST # 125 - WRITE AND READ 0'S TO ALL HIGH CACHE VALID
:*****
:*TEST 125 WRITE AND READ 0'S TO ALL HIGH CACHE VALID
:* WRITE AND READ 0'S TO ALL HIGH CACHE VALID
:* BIT STORE ADDRESS LOCATIONS- SET A
:* (VALID STORE LOCATIONS 4000 TO 7777)
:*****
TST1'S:
```

```
020324
020324 000004          SPCOND          :SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
                                :ERROR/LOOP ON TEST
020326 020336          .WORD 40$          :TEST START LOCATION
020330 060052          .WORD 1$-40$+57764 :LOOP ON ERROR START LOCATION
020332 000000          .WORD 0          :SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
020334 060100          .WORD 25$-40$+57764 :LOOP ON ERROR END LOCATION
020336 012737 001015 177746 40$: MOV #OFF,CCR :DISABLE CACHE
020344 004437 002424          JSR R4,RELCTL :LOCATE TEST CODE TO LOW CACHE SPACE
020350 020536          .WORD 10$+2       :ADDRESS OF START OF NEXT TEST
```

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

```
3078 020352 032737 020000 177746          BIT #VSIU,CCR :IS SET A BEING USED?
3079 020360 001407          BEQ 3$          :YES
3080 020362 052737 000400 177746          BIS #FC,CCR    :NO: FLUSH CACHE FOR SET A
3081 020370 032737 010000 177746 200$: BIT #VCIP,CCR  :WAIT TILL FLUSH COMPLETE
3082 020376 001374          BNE 200$
3083 020400 012737 177777 177752 3$: MOV #-1,CHR    :LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
3084                                     :REGISTERS, SINCE TDAR WILL BE USED
3085 020406 112737 000374 177751          MOVB #374,CMR+1
3086 020414 012705 070000          MOV #70000,R5  ;;ADDRESS 70000 INTO R5
3087 020420 012703 050000          MOV #50000,R3  :ADDRESS 50000 INTO R3
3088 020424 012737 000015 177746 1$: MOV #15,CCR   :NO UCB SO AS TO WRITE ENABLE VALID STORE
3089 020432 112737 000003 177750          MOVB #HODO+TDAR,CMR :HODO ALLOWS VALID STORE SET A TO
3090                                     :BE WRITTEN TO CMR<12> ONLY DURING
3091                                     :THE DESTINATION MEMORY ACCESS.
3092                                     :TDAR WILL FORCE A 0 TO BE WRITTEN
3093                                     :INTO VALID STORE WHEN A WRITE TO
3094                                     :VALID STORE OCCURS
3095 020440 005713          TST (R3)
3096 020442 005715          TST (R5)      ;WRITE A 0 INTO VALID STORE ADDRESS
3097                                     ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3098 020444 005715          TST (R5)      ;WRITE VALID STORE DATA INTO CMR<12>
3099                                     ;FROM VALID STORE ADDRESS LOCATION
3100                                     ;JUST WRITTEN INTO.
3101 020446 013701 177750          MOV CMR,R1    ;SAVE CMR DATA
3102 020452 000240          NOP          ;INSTRUCTION 'JMP 1$' PLACED HERE
3103 020454 000240          NOP          ;FOR LOOP ON ERROR
3104 020456 105037 177750          CLRB CMR     ;DISABLE MAINT. MODE
3105 020462 012737 001015 177746          MOV #OFF,CCR  ;DISABLE CACHE
3106 020470 032701 010000          BIT #VLD,R1  ;CMR<12> SHOULD BE 0.
3107 020474 001410          BEQ 9$
3108 020476 010537 050506          MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
3109 020502 006237 050506          ASR CA121    ;USED: CA<12:1>
3110 020506 104413          ERROR       ;PREPARE CA121 FOR TYPEOUT
3111                                     ;ERROR
3112                                     ;-----
020510 020506          .WORD -.2
```

```

3111
3112
3113
3114 020512 050506
3115
3116 020514 000000
3117 020516 062705 000002
3118 020522 062703 000002
3119 020526 020527 100000
3120 020532 001334
3121 020534 000240
      020536 00523 001472

```

```

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

```

```

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

```

WRITE AND READ 1'S TO ALL HIGH CACHE VALID

SBTTL TEST # 126 - WRITE AND READ 1'S TO ALL HIGH CACHE VALID

TEST 126 WRITE AND READ 1'S TO ALL HIGH CACHE VALID
WRITE AND READ 1'S TO ALL HIGH CACHE VALID
BIT STORE ADDRESS LOCATIONS- SET A
(VVALID STORE LOCATIONS 4000 TO 7777)

TST126: SCPCND ;SCOPE CONDITIONS:GO SFT 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

3128 020570 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
3129 020576 001407 BEQ 3\$;YES
3130 020600 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A
3131 020606 032737 010000 177746 200\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3132 020614 001374 BNE 200\$
3133 020616 012705 070000 3\$: MOV #70000,R5 ;:ADDRESS 70000 INTO R5
3134 020622 012703 050000 MOV #50000,R3 ;ADDRESS 50000 INTO R3
3135 020626 012737 000015 177746 1\$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3136 020634 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS VALID STORE SET A TO
;BE WRITTEN TO CMR<12> ONLY DURING
;THE DESTINATION MEMORY ACCESS.
3137
3138
3139 020642 005713 TST (R3)
3140 020644 005715 TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3141
3142 020646 005715 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
;FROM VALID STORE ADDRESS LOCATION
3143
3144 ;JUST WRITTEN INTO.
3145 020650 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
3146 020654 060240 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
020656 000240 NOP ;FOR LOOP ON ERROR
3147 020660 105037 177750 CLR B CMR ;DISABLE MAINT. MODE
3148 020664 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3149 020672 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 1.
3150 020676 001010 BNE 9\$;PASS
3151 020700 010537 050506 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
3152 ;USED: CA<12:1>
3153 020704 006237 050506 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3154 020710 104413 ERROR ;ERROR
;-----
020712 020710 .WORD -2
3155 ;VALID BITS STORE TESTS
3156 ;READING VALID STORE DATA SET A
3157 ;THRU CMR<12> DID NOT RESULT IN 0.
3158 020714 050506 CA121 ;PRINT VALID STORE ADDRESS LOCATION
3159 ;USED: CA<12:1>.
3160 020716 000000 .WORD 0

MACRO M1113 28-MAR-81 14:20 PAGE 108-1
TEST # 26 - WRITE AND READ TESTS TO ALL HIGH CACHE VALID

C 11

SEQUENCE 132

3161	020720	062705	000002
3162	020724	062703	000002
3163	020730	020527	100000
3164	020734	001334	
3165	020736	000240	
	020740	005237	001472

9S:	ADD	#2,R5
	ADD	#2,R3
	CMP	R5,#100000
	BNE	1S
10S:	NOP	
	INC	\$TESTN

:NEXT VALID STORE LOCATION
:HAVE ALL HIGH CACHE ADDRESS LOCATIONS BEEN DONE?
:NO
:END OF TEST
:INCREMENT TEST COUNTER

3169

```

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

```

```

020744
020744 000004          SCPCND          ;SCOPE CONDITIONS:GO SET JP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
020746 020756          .WORD 40$          ;TEST START LOCATION
020750 070054          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
020752 000000          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
020754 070130          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
020756 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
020764 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
020770 021206          .WORD 10$+2       ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

3170 020772 032737 020000 177746          BIT #VSIU,CCR ;IS SET A BEING USED?
3171 021000 001407          BEQ 3$          ;YES
3172 021002 052737 000400 177746          BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A
3173 021010 032737 010000 177746 4$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3174 021016 001374          BNE 4$
3175 021020 012737 000002 050516 3$: MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
3176 021026 012702 040000 2$: MOV #40000,R2 ;
3177 021032 012703 060000          MOV #60000,R3 ;
3178 021036 063702 050516          ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
3179 021042 063703 050516          ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
3180 021046 112737 000002 177750 1$: MOV#B #HODO,CMR ;HODO ALLOWS VALID DATA STORE BITS TO BE
3181          ;WRITTEN TO CMR<12> ONLY DURING THE
3182          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3183 021054 012737 000015 177746          MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3184 021062 152737 000001 177750          BISB #TDAR,CMR ;TDAR WILL FORCE A 0 TO BE WRITTEN
3185          ;INTO VALID STORE WHEN A WRITE TO
3186          ;VALID STORE OCCURS.
3187 021070 005737 040000          TST 40000
3188 021074 005737 060000          TST 60000 ;WRITE 0 INTO VALID STORE LOCATION 0000.
3189 021100 142737 000001 177750          BICB #TDAR,CMR ;CLEARING TDAR WILL ALLOW A 1 TO BE
3190          ;WRITTEN INTO VALID STORE WHEN A WRITE
3191          ;TO VALID STORE OCCURS.
3192 021106 005712          TST (R2)
3193 021110 005713          TST (R3) ;WRITE 1 INTO VALID STORE LOCATION
3194          ;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
3195 021112 005737 060000          TST 60000 ;LOAD DATA FROM VALID DATA STORE LOCATION
3196          ;0000 INTO CMR<12>.
3197 021116 013701 177750          MOV CMR,R1 ;SAVE CMR DATA
3198 021122 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
3199 021124 000240          NOP ;FOR LOOP ON ERROR
3199 021126 105037 177750          CLR#B CMR ;DISABLE MAINT. MODE
3200 021132 012737 001015 177746          MOV #OFF,CCR ;DISABLE CACHE
3201 021140 032701 010000          BIT #VLD,R1 ;CMR<12> SHOULD READ 0.
3202 021144 001411          BEQ 9$ ;PASS
3203          ;ROUTINE
3204 021146 013737 050516 050506          MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
3205 021154 006237 050506          ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3206 021160 104413          ERROR ;ERROR

```

```

021162 021160 .WORD -2 ;-----
3207 ;VALID STOR ADDRESS VERIFICATION.
3208 ;VALID STORE LOCATION 0000 DID NOT
3209 ;READ AS A 0 INDICATING THAT IT WAS
3210 ;OVERWRITTEN WITH A 1. THIS SUGGESTS
3211 ;A BAD CA<12:1> VALID STORE ADDRESS LINE.
021164 050506 CA'21 ;PRINT VALID STORE ADDRESS FAILURE. CA<12:1>.
3212 ;NOTE THAT THE 1 IN THIS PATTERN
3213 ;WILL POINT TO THE ADDRESS LINE OF
3214 ;THAT BROUGHT OUT ERROR.
3215
3216 021166 000000 .WORD 0
3217 021170 006337 050516 9$: ASL FLTPAT ;NEXT PATTERN
3218 021174 022737 020000 050516 CMP #20000,FLTPAT ;HAS VALID DATA STORE ADDRESS 4000 BEEN DONE?
3219 021202 001311 BNE 2$ ;NO
3220 021204 000240 10$: NOP ;END OF TEST
021206 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER

```

3226

```

.SBTTL TEST # 130 - LOW CACHE INVALIDATE WITH CACHE FLUSH
*****
*TEST 130      LOW CACHE INVALIDATE WITH CACHE FLUSH
*      VERIFY THAT ALL LOW CACHE VALID STORE SET A ADDRESS LOCATIONS
*      WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
*      (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 0000-5777)
*****
    
```

```

021212
021212 000004          SCPCND          ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
021214 021224          .WORD 40$          ;LOOP ON ERROR START LOCATION
021216 070074          .WORD 1$-40$+67764      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
021220 000000          .WORD 0          ;LOOP ON ERROR END LOCATION
021222 070130          .WORD 25$-40$+67764
021224 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
021232 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
021236 021464          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST
    
```

```

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

```

3227 021240 012705 060000          MOV #60000,R5 ;:ADDRESS 60000 INTO R5
3228 021244 012703 040000          MOV #40000,R3 ;:ADDRESS 40000 INTO R3
3229 021250 032737 020000 177746 BIT #VSIU,CCR ;:IS SET A BEING USED?
3230 021256 001407          BEQ 3$ ;:YES
3231 021260 052737 000400 177746 BIS #FC,CCR ;:NO; FLUSH CACHE FOR SET A
3232 021266 032737 010000 177746 200$: BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
3233 021274 001374          BNE 200$
3234 021276 012737 000015 177746 3$: MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
3235 021304 112737 000002 177750 MOV#B #HODO,CMR ;:HODO ALLOWS VALID STORE SET A TO
3236 ;:BE WRITTEN TO CMR<12> ONLY DURING
3237 ;:THE DESTINATION MEMORY ACCESS.
3238 021312 005713          4$: TST (R3)
3239 021314 005715          TST (R5) ;:WRITE A 1 INTO VALID STORE ADDRESS
3240 ;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3241 021316 062705 000002          ADD #2,R5 ;:NEXT VALID STORE LOCATION
3242 021322 062703 000002          ADD #2,R3
3243 021326 020527 070000          CMP R5,#70000 ;:HAVE ALL LOW CACHE LOCATIONS BEEN DONE?
3244 021332 001367          BNE 4$ ;:NO
3245 021334 052737 000400 177746 1$: BIS #FC,CCR ;:FLUSH CACHE TO SELECT SET R AND
3246 ;:INVALIDATE SET A
3247 021342 032737 010000 177746 500$: BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
3248 021350 001374          BNE 500$
3249 021352 052737 000400 177746 BIS #FC,CCR ;:FLUSH TO SELECT SET A AGAIN
3250 021360 032737 010000 177746 6$: BIT #VCIP,CCR ;:WAIT
3251 021366 001374          BNE 6$
3252 021370 000240          25$: NOP ;:INSTRUCTION 'JMP 1$' PLACED HERE
3253 021372 000240          NOP ;:FOR LOOP ON ERROR
3254 021374 012705 060000          2$: MOV #60000,R5 ;:ADDRESS 60000 INTO R5
3255 021400 005715          TST (R5) ;:WRITE VALID STORE DATA INTO CMR<12>
3256 ;:FROM ADDRESS LOCATION SPECIFIED BY
3257 ;:R5'S BITS 12-1.
3257 021402 013701 177750          MOV CMR,R1 ;:SAVE CMR DATA
3258 021406 032701 010000          BIT #VLD,R1 ;:CMR<12> SHOULD BE 0
3259 021412 001416          BEQ 9$ ;:PASS
3260 021414 105037 177750          CLRB CMR ;:DISABLE MAINT. MODE
3261 021420 012737 001015 177746 MOV #OFF,CCR ;:DISABLE CACHE
    
```


3284

```
.SBTTL TEST # 131 - HIGH CACHE INVALIDATE WITH CACHE FLUSH
*****
*TEST 131 HIGH CACHE INVALIDATE WITH CACHE FLUSH
*   VERIFY THAT ALL HI CACHE VALID STORE SET A ADDRESS LOCATIONS
*   WILL BE INVALIDATED AS A RESULT OF A CACHE FLUSH.
*   (FLUSH COUNTER ADDRESS LOCATIONS [CNT<12:1>]: 4000-7777)
*****
TST131:
```

```
021470
021470 000004

021472 021502
021474 060074
021476 000000
021500 060130
021502 012737
021510 004437
021514 021742
```

```
001015 177746 40$: SCPCND ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
002424 ;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCCPE 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
```

```
3285 021516 012705 070000 MOV #70000,R5 ;:ADDRESS 70000 INTO R5
3286 021522 012703 050000 MOV #50000,R3 ;:ADDRESS 50000 INTO R3
3287 021526 032737 020000 177746 BIT #VSIU,CCR ;:IS SET A BEING USED?
3288 021534 001407 BEQ 3$ ;:YES
3289 021536 052737 000400 177746 BIS #FC,CCR ;:NO; FLUSH CACHE FOR SET A
3290 021544 032737 010000 177746 200$: BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
3291 021552 001374 BNE 200$
3292 021554 012737 000015 177746 3$: MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
3293 021562 112737 000002 177750 MOVB #HODO,CMR ;:HODO ALLOWS VALID STORE SET A TO
;:BE WRITTEN TO CMR<12> ONLY DURING
;:THE DESTINATION MEMORY ACCESS.
3294
3295
3296 021570 005713 4$: TST (R3)
3297 021572 005715 TST (R5) ;:WRITE A 1 INTO VALID STORE ADDRESS
;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
;:NEXT VALID STORE LOCATION
3298
3299 021574 062705 000002 ADD #2,R5
3300 021600 062703 000002 ADD #2,R3
3301 021604 020527 100000 CMP R5,#100000 ;:HAVE ALL HI CACHE LOCATIONS BEEN DONE?
3302 021610 001367 BNE 4$ ;:NO
3303 021612 052737 000400 177746 1$: BIS #FC,CCR ;:FLUSH CACHE TO SELECT SET B AND
;:INVALIDATE SET A
3304 ;:WAIT TILL FLUSH COMPLETE
3305 021620 032737 010000 177746 500$: BIT #VCIP,CCR
3306 021626 001374 BNE 500$
3307 021630 052737 000400 177746 BIS #FC,CCR ;:FLUSH TO SELECT SET A AGAIN
3308 021636 032737 010000 177746 6$: BIT #VCIP,CCR ;:WAIT
3309 021644 001374 BNE 6$
3310 021646 000240 25$: NOP ;:INSTRUCTION 'JMP 1$' PLACED HERE
;:FOR LOOP ON ERROR
021650 000240 NOP ;:ADDRESS 70000 INTO R5
3311 021652 012705 070000 MOV #70000,R5 ;:WRITE VALID STORE DATA INTO CMR<12>
3312 021656 005715 2$: TST (R5) ;:FROM ADDRESS LOCATION SPECIFIED BY
;:R5'S BITS 12-1.
3313
3314 ;:SAVE CMR DATA
3315 021660 013701 177750 MOV CMR,R1 ;:CMR<12> SHOULD BE 0
3316 021664 032701 010000 BIT #VLD,R1 ;:PASS
3317 021670 001416 BEQ 9$ ;:DISABLE MAINT. MODE
3318 021672 105037 177750 CLRB CMR
3319 021676 012737 001015 177746 MOV #OFF,CCR ;:DISABLE CACHE
```

HI-CACHE INVALIDATE WITH CACHE FLUSH

```

3320 021704 010537 050514      MOV      R5,CNT121      ;SAVE VALID STORE FLUSH ADDRESS LOCATION
3321 021704 000000 000000      ;USED: CNT<12:1>
3322 021710 006237 050514      ASR      CNT121        ;PREPARE CNT121 FOR TYPEOUT
3323 021714 104613 000000      ERROR    ;ERROR
3324 021716 021714 000000      .WORD   .-2           ;-----
3325 021716 021714 000000      ;FLUSH CACHE TEST-SET A
3326 021716 021714 000000      ;READING VALID STORE LOCATION FROM SET A THRU (MR-1)
3327 021716 021714 000000      ;DID NOT RESULT IN A ZERO,INDICATING THAT
3328 021716 021714 000000      ;THE CACHE FLUSH DID NOT INVALIDATE THIS
3329 021720 050514 000000      CNT121  ;LOCATION.
3330 021720 050514 000000      ;PRINT VALID STORE FLUSH ADDRESS LOCATION
3331 021722 000000 000000      .WORD   0             ;IN ERROR: CNT<12:1>.
3332 021724 000405 000000      BR       10$          ;IF ERROR, END TEST
3333 021726 062705 000002      ADD     #2,R5         ;NEXT VALID STORE LOCATION
3334 021732 020527 100000      CMP     R5,#100000    ;HAVE ALL HI CACHE ADDRESS LOCATIONS BEEN DONE?
3335 021736 001347 000000      BNE     2$            ;NO
3336 021740 000240 001472      NOP     ;END OF TEST
3337 021742 005237 0C1472      INC     $TESTN        ;INCREMENT TEST COUNTER

```

3342

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

```
021746
021746 000004

021750 021760
021752 070052
021754 000000
021756 070100
021760 012737 001015 177746
021766 004437 002452
021772 022160
```

```
TST132:
SCPCND ;SCOPE CONDITIONS:GO SFT 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

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

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

```
3343 021774 032737 020000 177746
3344 022002 001007
3345 022004 052737 000400 177746
3346 022012 032737 010000 177746
3347 022020 001374
3348 022022 012737 177777 177752
3349
3350 022030 112737 000374 177751
3351 022036 012705 060000
3352 022042 012703 040000
3353 022046 012737 000015 177746
3354 022054 112737 000003 177750
3355
3356
3357
3358
3359
3360 022062 005713
3361 022064 005715
3362
3363 022066 005715
3364
3365
3366 022070 013701 177750
3367 022074 000240
022076 000240
3368 022100 105037 177750
3369 022104 012737 001015 177746
3370 022112 032701 010000
3371 022116 001410
3372 022120 010537 050506
3373
3374 022124 006237 050506
3375 022130 104413

022132 022130
```

```
BIT #VSIU,CCR ;IS SET B BEING USED?
BNE 3$ ;YES
BIS #FC,CCR ;NO: FLUSH CACHE FOR SET B
BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
BNE 200$
MOV #-1,CHR ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
;REGISTERS, SINCE TDAR WILL BE USED

MOV #374,CMR+1
MOV #60000,R5 ;;ADDRESS 60000 INTO R5
MOV #40000,R3 ;;ADDRESS 40000 INTO R3
MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
MOV #HODU+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

TST (R3)
TST (R5) ;WRITE A 0 INTO VALID STORE ADDRESS
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
;FROM VALID STORE ADDRESS LOCATION
;JUST WRITTEN INTO.
MOV CMR,R1 ;SAVE CMR DATA
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
BIT #VLD,R1 ;CMR<12> SHOULD BE 0.
BEQ 9$ ;PASS
MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
;USED: CA<12:1>
ASR CA121 ;PREPARE CA121 FOR TYPEOUT
ERROR ;ERROR
;-----

.WORD -2
```

3376
3377
3378
3379 022134 050506
3380
3381 022136 000000
3382 022140 062705 000002
3383 022144 062703 000002
3384 022150 020527 070000
3385 022154 001334
3386 022156 000240
022160 005237 001472

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

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

:NEXT VALID STORE LOCATION

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

3392

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

022164
022164 000004

022166 022176
022170 070036
022172 000000
022174 070064
022176 012737 001015 177746
022204 004437 002452
022210 022362

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

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

3393 022212 032737 020000 177746
3394 022220 001007
3395 022222 052737 000400 177746
3396 022230 032737 010000 177746
3397 022236 001374
3398 022240 012705 060000
3399 022244 012703 040000
3400 022250 012737 000015 177746
3401 022256 112737 000002 177750
3402
3403
3404 022264 005713
3405 022266 005715
3406
3407 022270 005715
3408
3409
3410 022272 013701 177750
3411 022276 000240
022300 000240
3412 022302 105037 177750
3413 022306 012737 001015 177746
3414 022314 032701 010000
3415 022320 001010
3416
3417 022322 010537 050506
3418
3419 022326 006237 050506
3420 022332 104413

022334 022332
3421
3422
3423
3424 022336 050506
3425

```
BIT #VSIU,CCR ;IS SET B BEING USED?
BNE 3$ ;YES
BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B
200$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
BNE 200$
3$: MOV #60000,R5 ;;ADDRESS 60000 INTO R5
MOV #40000,R3 ;;ADDRESS 40000 INTO R3
1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
MOV# #HODO,CMR ;HODO ALLOWS VALID STORE SET B TO
;BE WRITTEN TO CMR<12> ONLY DURING
;THE DESTINATION MEMORY ACCESS.
TST (R3)
TST (R5) ;WRITE A 1 INTO VALID STORE ADDRESS
;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
;FROM VALID STORE ADDRESS LOCATION
;JUST WRITTEN INTO.
MOV CMR,R1 ;SAVE CMR DATA
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
BIT #VLD,R1 ;CMR<12> SHOULD BE 1.
BNE 9$ ;PASS
MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
;USED: CA<12:1>
3419 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3420 ERROR ;ERROR
;-----
.WORD -2
;VALID BITS STORE TESTS - SET B
;READING VALID STORE DATA SET B
;THRU CMR<12> DID NOT RESULT IN 1.
CA121 ;PRINT VALID STORE ADDRESS LOCATION
;USED: CA<12:1>.
```

3426	022340	000000		.WORD	0	
3427	022342	062705	000002	9\$: ADD	#2,R5	:NEXT VALID STORE LOCATION
3428	022346	062703	000002	ADD	#2,R3	
3429	022352	020527	070000	CMP	R5,#70000	:HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE?
3430	022356	001334		BNE	1\$:NO
3431	022360	000240		10\$: NOP		:END OF TEST
	022362	005237	001472	INC	\$TES*N	:INCREMENT TEST COUNTER

3437

```

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

```

TST134:
022366 000004          SCPCND          ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
022370 022400          .WORD 40$          ;ERROR/LOOP ON TEST
022372 060052          .WORD 1$-40$+57764 ;TEST START LOCATION
022374 000000          .WORD 0            ;LOOP ON ERROR START LOCATION
022376 060100          .WORD 25$-40$+57764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
022400 012737 001015 177746 40$: MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
022406 004437 002424 JSR R4,RELCTL ;DISABLE CACHE
022412 022600          .WORD 10$+2        ;LOCATE TEST CODE TO LOW CACHE SPACE
                                ;ADDRESS OF START OF NEXT TEST
  
```

```

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

```

3438 022414 032737 020000 177746 BIT #VSIU,CCR ;IS SET B BEING USED?
3439 022422 001007 BNE 3$ ;YES
3440 022424 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B
3441 022432 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3442 022440 001374 BNE 200$
3443 022442 012737 177777 177752 3$: MOV #-1,CHR ;LOAD AMR<21:0> WITH 1'S VIA CHR AND CMR
3444          ;REGISTERS, SINCE TDAR WILL BE USED
3445 022450 112737 000374 177751 MOVB #374,CMR+1
3446 022456 012705 070000 MOV #70000,R5 ;:ADDRESS 70000 INTO R5
3447 022462 012703 050000 MOV #50000,R3 ;:ADDRESS 50000 INTO R3
3448 022466 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
3449 022474 112737 000003 177750 MOVB #HODO+TDAR,CMR ;HODO ALLOWS VALID STORE SET B TO
3450          ;BE WRITTEN TO CMR<12> ONLY DURING
3451          ;THE DESTINATION MEMORY ACCESS.
3452          ;TDAR WILL FORCE A 0 TO BE WRITTEN
3453          ;INTO VALID STORE WHEN A WRITE TO
3454          ;VALID STORE OCCURS
3455 022502 005713 TST (R3)
3456 022504 005715 TST (R5) ;WRITE A 0 INTO VALID STORE ADDRESS
3457          ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
3458 022506 005715 TST (R5) ;WRITE VALID STORE DATA INTO CMR<12>
3459          ;FROM VALID STORE ADDRESS LOCATION
3460          ;JUST WRITTEN INTO.
3461 022510 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
3462 022514 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
022516 000240 NOP ;FOR LOOP ON ERROR
3463 022520 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
3464 022524 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3465 022532 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD BE 0.
3466 022536 001410 BEQ 9$ ;PASS
3467
3468 022540 010537 050506 MOV R5,CA121 ;SAVE VALID STORE ADDRESS LOCATION
3469          ;USED: CA<12:1>
3470 022544 006237 050506 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
3471 022550 104413 ERROR ;ERROR
          ;-----
  
```

MACRO M113 28-MAR-81 14:20 PAGE 114-1

WRITE AND READ OPS TO ALL HIGH CACHE VALID

```

3471 022552 022550
3472
3473
3474
3475 022554 050506
3476
3477 022556 000000
3478 022560 062705 000002
3479 022564 062703 000002
3480 022570 020527 100000
3481 022574 001334
3482 022576 000240
      022600 005237 0014,2

```

```

      .WORD    -2
      CA12?
      .WORD    0
9%:   ADD     #2,R5
      ADD     #2,R3
      CMP     R5,#100000
      BNE    1$
10%:  NOP
      INC    $TESTN

```

```

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

```


3488

```
.SBTTL TEST # 135 - WRITE AND READ 1'S TO ALL HIGH CACHE VALID
*****
*TEST 135 WRITE AND READ 1'S TO ALL HIGH CACHE VALID
* WRITE AND READ 1'S TO ALL HIGH CACHE VALID
* BIT STORE ADDRESS LOCATIONS- SET B
* (VALID STORE LOCATIONS 4000 TO 7777)
*****
TST135:
```

022604
022604 000004

022606 022616
022610 060036
022612 000000
022614 060064
022616 012737 001015 177746
022624 004437 002424
022630 023002

```
SCPCND ;SCOPE CONDITIONS:GO SFT 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
```

3489 022632 032737 020000 177746
3490 022640 001007
3491 022642 052737 000400 177746
3492 022650 032737 010000 177746
3493 022656 001374
3494 022660 012705 070000
3495 022664 012703 050000
3496 022670 012737 000015 177746
3497 022676 112737 000002 177750
3498
3499
3500 022704 005713
3501 022706 005715
3502
3503 022710 005715
3504
3505
3506 022712 013701 177750
3507 022716 000240
022720 000240
3508 022722 105037 177750
3509 022726 012737 001015 177746
3510 022734 032701 010000
3511 022740 001010
3512 022742 010537 050506
3513
3514 022746 006237 050506
3515 022752 104413

022754 022752
3516
3517
3518
3519 022756 050506
3520
3521 022760 000000

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

3522	022762	062705	000002	9\$:	ADD	#2,R5	:NEXT VALID STORE LOCATION
3523	022766	062703	000002		ADD	#2,R3	
3524	022772	020527	100000		CMP	R5,#100000	:HAVE ALL HIGH CACHE ADDRESS LOCATIONS BEEN DONE?
3525	022776	001334			BNE	1\$:NO
3526	023000	000240		10\$:	NOP		:END OF TEST
	023002	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

3530

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

 *TEST 136 CHK VALID DATA STORE ADRS LINES (CA(12:1)) SET B
 * VERIFY VALID DATA STORE ADDRESS LINES (CA(12:1)) - SET B

TST136:
 023006 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 023006 000004 ;ERROR/LOOP ON TEST
 023010 023020 .WORD 40\$;TEST START LOCATION
 023012 070054 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION
 023014 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
 023016 070130 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION
 023020 012737 001015 177746 40\$: MOV #OFF,CCR ;DISABLE CACHE
 023026 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
 023032 023250 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

3531 023034 032737 020000 177746 BIT #VSIU,CCR ;IS SET B BEING USED?
 3532 023042 001007 BNE 3\$;YES
 3533 023044 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET B
 3534 023052 032737 010000 177746 4\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
 3535 023060 001374 BNE 4\$
 3536 023062 012737 000002 050516 3\$: MOV #2,FLTPAT ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
 3537 023070 012702 040000 2\$: MOV #40000,R2 ;
 3538 023074 012703 060000 MOV #60000,R3 ;
 3539 023100 063702 050516 ADD FLTPAT,R2 ;R2 CONTAINS 40000+FLTPAT
 3540 023104 063703 050516 ADD FLTPAT,R3 ;R3 CONTAINS 60000+FLTPAT
 3541 023110 112737 000002 177750 1\$: MOVB #HODO,CMR ;HODO ALLOWS VALID DATA STORE BITS TO BE
 3542 ;WRITTEN TO CMR<12> ONLY DURING THE
 3543 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
 3544 023116 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE VALID STORE
 3545 023124 152737 000001 177750 BISB #TDAR,CMR ;TDAR WILL FORCE A 0 TO BE WRITTEN
 3546 ;INTO VALID STORE WHEN A WRITE TO
 3547 ;VALID STORE OCCURS.
 3548 023132 005737 040000 TST 40000
 3549 023136 005737 060000 TST 60000 ;WRITE 0 INTO VALID STORE LOCATION 0000.
 3550 023142 142737 000001 177750 BICB #TDAR,CMR ;CLEARING TDAR WILL ALLOW A 1 TO BE
 3551 ;WRITTEN INTO VALID STORE WHEN A WRITE
 3552 ;TO VALID STORE OCCURS.
 3553 023150 005712 TST (R2)
 3554 023152 005713 TST (R3) ;WRITE 1 INTO VALID STORE LOCATION
 3555 ;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
 3556 023154 005737 060000 TST 60000 ;LOAD DATA FROM VALID DATA STORE LOCATION
 3557 ;0000 INTO CMR<12>.
 3558 023160 013701 177750 MOV CMR,R1 ;SAVE CMR DATA
 3559 023164 000240 25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
 023166 000240 NOP ;FOR LOOP ON ERROR
 3560 023170 105037 177750 CLRB CMR ;DISABLE MAINT. MODE
 3561 023174 012737 0C1015 177746 MOV #OFF,CCR ;DISABLE CACHE
 3562 023202 032701 010000 BIT #VLD,R1 ;CMR<12> SHOULD READ 0.
 3563 023206 001411 BEQ 9\$;PASS
 3564 ;ROUTINE
 3565 023210 013737 050516 050506 MOV FLTPAT,CA121 ;SAVE CA<12:1> USED
 3566 023216 006237 050506 ASR CA121 ;PREPARE CA121 FOR TYPEOUT
 3567 023222 104413 ERROR ;ERROR

3568 023224 023222

.WORD .-2

:-----

3569
3570
3571
3572
3573
3574
3575
3576

023226 050506

(A121

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

3577 023230 000000

.WORD 0

3578 023232 006337 050516 9\$:

ASL FLTPAT

:NEXT PATTERN

3579 023236 022737 020000 050516

CMP #20000,FLTPAT

:HAS VALID DATA STORE ADDRESS 4000 BEEN DONE?

3580 023244 001311

BNE 2\$

:NO

3581 023246 000240

10\$:

NOP

:END OF TEST

023250 005237 001472

INC \$TESTN

:INCREMENT TEST COUNTER

ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS

SBTTL TEST # 137 - ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS

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

023254 000004

023256 023266
023260 070074
023262 000000
023264 070130
023266 012737
023274 004437
023300 023526

001015 177746
002452

TST137:
SCPCND ;SCOPE CONDITIONS:GO SFT 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

3588 023302 012705 060000
3589 023306 012703 040000
3590 023312 032737 020000 177746
3591 023320 001007
3592 023322 052737 000400 177746
3593 023330 032737 010000 177746 200\$:
3594 023336 001374
3595 023340 012737 000015 177746 3\$:
3596 023346 112737 000002 177750
3597
3598
3599 023354 005713 4\$:
3600 023356 005715
3601
3602 023360 062705 000002
3603 023364 062703 000002
3604 023370 020527 070000
3605 023374 001367
3606 023376 052737 000400 177746 1\$:
3607
3608 023404 032737 010000 177746 500\$:
3609 023412 001374
3610 023414 052737 000400 177746
3611 023422 032737 010000 177746 6\$:
3612 023430 001374
3613 023432 000240 25\$:
3614 023434 000240
3615 023436 012705 060000 2\$:
3616
3617
3618 023444 013701 177750
3619 023450 032701 010000
3620 023454 001416
3621 023456 105037 177750
3622 023462 012737 001015 177746

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

TEST # 137 - ALL LOW CACHE VALID STORE SET B ADDRESS LOCATIONS

3623	023470	010537	050514	MOV	R5,CNT121			:SAVE VALID STORE FLUSH ADDRESS LOCATION
3624								:USED: CNT<12:1>
3625	023474	006237	050514	ASR	CNT121			:PREPARE CNT121 FOR TYPEOUT
3626	023500	104413		ERROR				:ERROR
								:-----
	023502	023500		.WORD	.-2			:FLUSH CACHE INVALID TEST-SET B
3627								:READING VALID STORE LOCATION FROM SET B THRU (MR. 12)
3628								:DID NOT RESULT IN A ZERO,INDICATING THAT
3629								:THE CACHE FLUSH DID NOT INVALIDATE THIS
3630								:LOCATION.
3631								:PRINT VALID STORE FLUSH ADDRESS LOCATION
3632	027504	050514		CNT121				:IN ERROR: CNT<12:1>.
3633								
3634	023506	000000		.WORD	0			
3635	023510	000405		BR	10\$:IF ERROR, END TEST
3636	023512	062705	000002	9\$:	ADD	#2,R5		:NEXT VALID STORE LOCATION
3637	023516	020527	070000	10\$:	CMP	R5,#70000		:HAVE ALL LOW CACHE ADDRESS LOCATIONS BEEN DONE:
3638	023522	001347			BNE	2\$:NO
3639	023524	000240			NOP			:END OF TEST
	023526	005237	001472		INC	\$TESTN		:INCREMENT TEST COUNTER

TEST

SBTTL TEST # 140 - ALL HI CACHE VALID STORE SET B ADDRESS LOCATIONS

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

023532
 023532 000004

TEST140:

023534 023544
 023536 060074
 023540 000000
 023542 060130
 023544 012737 001015 177746
 023552 004437 002424
 023556 024004

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

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

3646 023560 012705 070000
 3647 023564 012703 050000
 3648 023570 032737 020000 177746
 3649 023576 001007
 3650 023600 052737 000400 177746
 3651 023606 032737 010000 177746 200\$:
 3652 023614 001374
 3653 023616 012737 000015 177746 3\$:
 3654 023624 112737 000002 177750
 3655
 3656
 3657 023632 005713 4\$:
 3658 023634 005715
 3659
 3660 023636 062705 000002
 3661 023642 062703 000002
 3662 023646 020527 100000
 3663 023652 001367
 3664 023654 052737 000400 177746 1\$:
 3665
 3666 023662 032737 010000 177746 500\$:
 3667 023670 001374
 3668 023672 052737 000400 177746
 3669 023700 032737 010000 177746 6\$:
 3670 023706 001374
 3671 023710 000240 25\$:
 023712 000240
 3672 023714 012705 070000
 3673 023720 005715 2\$:
 3674
 3675
 3676 023722 013701 177750
 3677 023726 032701 010000
 3678 023732 001416
 3679 023734 105037 177750
 3680 023740 012737 001015 177746

MOV #70000,R5 ;:ADDRESS 70000 INTO R5
 MOV #50000,R3 ;:ADDRESS 50000 INTO R3
 BIT #VSIU,CCR ;:IS SET B BEING USED?
 BNE 3\$;:YES
 BIS #FC,CCR ;:NO; FLUSH CACHE FOR SET B
 BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
 BNE 200\$
 MOV #15,CCR ;:NO UCB SO AS TO WRITE ENABLE VALID STORE
 MOVB #HCDO,CMR ;:HODO ALLOWS VALID STORE SET B TO
 ;:BE WRITTEN TO CMR<12> ONLY DURING
 ;:THE DESTINATION MEMORY ACCESS.
 TST (R3) 4\$:
 TST (R5) ;:WRITE A 1 INTO VALID STORE ADDRESS
 ;:LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
 ADD #2,R5 ;:NEXT VALID STORE LOCATION
 ADD #2,R3
 CMP R5,#100000 ;:HAVE ALL HI CACHE LOCATIONS BEEN DONE?
 BNE 4\$;:NO
 BIS #FC,CCR ;:FLUSH CACHE TO SELECT SET A AND
 ;:INVALIDATE SET B
 BIT #VCIP,CCR ;:WAIT TILL FLUSH COMPLETE
 BNE 500\$
 BIS #FC,CCR ;:FLUSH TO SELECT SET B AGAIN
 BIT #VCIP,CCR 6\$:
 BNE 6\$;:WAIT
 NOP 25\$:
 NOP ;:INSTRUCTION 'JMP 1\$' PLACED HERE
 ;:FOR LOOP ON ERROR
 MOV #70000,R5 ;:ADDRESS 70000 INTO R5
 TST (R5) 2\$:
 ;:WRITE VALID STORE DATA INTO CMR<12>
 ;:FROM ADDRESS LOCATION SPECIFIED BY
 ;:R5'S BITS 12-1.
 MOV CMR,R1 ;:SAVE CMR DATA
 BIT #VLD,R1 ;:CMR<12> SHOULD BE 0
 BEQ 9\$;:PASS
 CLRB CMR ;:DISABLE MAINT MODE
 MOV #OFF,CCR ;:DISABLE CACHE

```

3681 023746 010537 050514      MOV      R5,CNT121      ;SAVE VALID STORE FLUSH ADDRESS LOCATION
3682                               ;USED: CNT<12:1>
3683 023752 006237 050514      ASR      CNT121        ;PREPARE CNT121 FOR TYPEOUT
3684 023756 104413              ERROR          ;ERROR
                               ;-----
                               023760 023756      .WORD     -2          ;FLUSH CACHE INVALID TEST-SET B
3685                               ;READING VALID STORE LOCATION FROM SET B THRU (MR<1?>
3686                               ;DID NOT RESULT IN A ZERO,INDICATING THAT
3687                               ;THE CACHE FLUSH DID NOT INVALIDATE THIS
3688                               ; LOCATION.
3689                               ;PRINT VALID STORE FLUSH ADDRESS LOCATION
3690 023762 050514      CNT121        ;!' ERROR: CNT<12:1>.
3691                               ;
3692 023764 000000      .WORD     0          ;IF ERROR,END TEST
3693 023766 000405      BR       10$        ;NEXT VALID STORE LOCATION
3694 023770 062705 000002      9$: ADD    #2,R5     ;HAVE ALL HI CACHE ADDRESS LOCATIONS BEEN DONE?
3695 023774 020527 100000      CMP     R5,#100000  ;NO
3696 024000 001347      BNE     2$          ;END OF TEST
3697 024002 000240      10$: NOP          ;INCREMENT TEST COUNTER
      024004 005237 004472      INC     $TESTN
    
```


373

```

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

```

```

024010
024010 000004          SCPCND          ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
024012 024022          .WORD 40$          ;TEST START LOCATION
024014 070000          .WORD 1$-40$+67764      ;LOOP ON ERROR START LOCATION
024016 000000          .WORD 0           ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
024020 070064          .WORD 25$-40$+67764  ;LOOP ON ERROR END LOCATION
024022 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
024030 004437 002452     JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
024034 024230          .WORD 10$+2       ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

3704 024036 005037 060000 1$: CLR 60000 ;0'S TO MAIN MEMORY LOCATION
3705 024042 005037 000000 CLR 0 ;CLEAR LOCATION 0
3706 024046 012700 177777 MOV #-1,R0 ;ALL 1'S TO R0
3707 024052 012701 060000 MOV #60000,R1 ;ADDRESS 60000 TO R1
3708 024056 112737 000002 177750 MOVB #HODO,CMR ;ALLOWS CACHE UPDATES & DATA STORE BITS TO BE
3709 ;WRITTEN TO CDR<15:0> ONLY DURING THE
3710 ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
3711 024064 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3712 024072 005737 060000 TST 60000 ;
3713 024076 005737 000000 TST 0 ;
3714 ;READ UPDATE; ALL 0'S TO DATA STORE
3715 ;LOCATION 0000 FROM MAIN MEMORY
3716 024102 010011 MOV R0,(R1) ;LOC. 0
3717 ;WRITE MISS:NO UPDATE SHOULD OCCUR
3718 024104 005711 TST (R1) ;TO DATA STORE LOCATION 0000
3719 ;READ MISS;LOAD DATA STORE BITS RESULTING
3720 024106 013702 177754 MOV CDR,R2 ;FROM PREVIOUS WRITE MISS INTO CDR<15:0>
3721 024112 010011 MOV R0,(R1) ;SAVE CDR CONTENTS
3722 ;WRITE HIT;
3723 ;THIS WRITE HIT SHOULD UPDATE DATA
3724 024114 005711 TST (R1) ;STORE LOCATION 0000.
3725 ;READ HIT;LOAD DATA STORE BITS RESULTING
3726 024116 013703 177754 MOV CDR,R3 ;FROM PREVIOUS WRITE HIT INTO CDR<15:0>
3727 024122 000240 25$: NOP ;SAVE CDR CONTENTS
024124 000240 NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
3728 024126 105037 177750 CLR CMR ;FOR LOOP ON ERROR
3729 024132 012737 001015 177746 MOV #OFF,CCR ;DISABLE MAINTENANCE
3730 024140 012737 000002 000000 MOV #2,0 ;DISABLE CACHE
3731 024146 005702 TST R2 ;RESTORE VECTOR
3732 024150 001411 BEQ 8$ ;CHECK FOR ALL 0'S
3733 024152 005037 050504 CLR EXDAT6 ;PASS
3734 024156 010237 050502 MOV R2,CDR150 ;SPECIFY EXPECTED DATA
3735 024162 104413 ERROR ;GET RECEIVED DATA FROM R2
                                ;ERROR
                                ;-----
024164 024162 .WORD -.2 ;WRITE CONTROL LOGIC TEST
3736

```

3737										:READING CDR<15:0> DID NOT RESULT IN ALL 0'S.
3738										:INDICATES THAT UPDATE OCCURED
3739										:DUE TO WRITE MISS.
3740	024166	050504				EXDAT6				:PRINT CDR<15:0> EXPECTED DATA
3741	024170	050502				CDR150				:PRINT CDR<15:0> DATA RECEIVED.
3742	024172	000000				.WORD	0			
3743	024174	022703	177777		8\$:	CMP	#-1,R3			:CHECK FOR ALL 1'S
3744	024200	001412				BEQ	10\$:PASS
3745	024202	012737	177777	050504		MOV	#-1,EXDAT6			:SPECIFY EXPECTED DATA
3746	024210	010337	050502			MOV	R3,CDR150			:GET RECEIVED DATA FROM R3
3747	024214	104413				ERROR				:ERROR
										:-----
	024216	024214				.WORD	.-2			:WRITE CONTROL LOGIC TEST
3748										:READING CDR<15:0> DID NOT RESULT IN ALL 1'S.
3749										:INDICATES THAT UPDATE DID NOT OCCUR
3750										:DUE TO WRITE HIT.
3751										:PRINT CDR<15:0> EXPECTED DATA
3752	024220	050504				EXDAT6				:PRINT CDR<15:0> DATA RECEIVED.
3753	024222	050502				CDR150				
3754	024224	000000				.WORD	0			
3755	024226	000240			10\$:	NOP				:END OF TEST
	024230	005237	001472			INC	\$TESTN			:INCREMENT TEST COUNTER

3763

```

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

```

```

024234
024234 000004

024236 024246
024240 070000
024242 000000
024244 070064
024246 012737 001015 177746
024254 004437 002452
024260 024440

```

```

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

```

```

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

```

```

3764 024262 012737 177777 177752 1$: MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING ALL 1'S TO CHR<8:0>
3765 024270 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE TAG FIELD BITS TO BE
3766 ;WRITTEN TO CHR<15:07> ONLY DURING
3767 ;THE DESTINATION MEMORY ACCESS
3768 ;OF AN INSTRUCTION
3769 024276 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE
3770 024304 005737 040000 TST 40000
3771 024310 005737 000000 TST 0 ;READ UPDATE;LOAD TAG STORE WITH ALL 0'S
3772 024314 052737 000001 177750 BIS #TDAR,CMR ;TDAR WILL ALLOW TAG STORE TO BE
3773 ;WRITTEN WITH CONTENTS OF AMR<8:0>
3774 ;IF AN UPDATE OCCURS.
3775 024322 005737 000000 TST 0 ;READ HIT; WRITE CONTROL LOGIC SHOULD
3776 ;BE INHIBITED FROM ISSUING A WRITE
3777 ;SIGNAL
3778 024326 005737 000000 TST 0 ;WRITE TAG FIELD DATA FROM TAG STORE
3779 ;LOCATION 0000 INTO CHR.
3780 024332 013737 177752 050474 MOV CHR,CHR157 ;SAVE CHR DATA
3781 024340 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3782 024346 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
024350 000240 NOP ;FOR LOOP ON ERROR
3783 024352 105037 177750 CLR B CMR ;DISABLE MAINTENANCE MODE
3784 024356 042737 000177 050474 BIC #177,CHR157 ;INTERESTED IN 15:07
3785 024364 005737 050474 TST CHR157 ;BITS 15:07 SHOULD BE ALL 0'S
3786 024370 001422 BEQ 10$ ;PASS
3787 024372 012737 000007 002062 MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
3788 024400 006237 050474 2$: ASP CHR157
3789 024404 042737 100000 050474 BIC #100000,CHR157
3790 024412 005337 002062 DEC LOOP
3791 024416 001370 BNE 2$
3792 024420 005037 050522 CLR EXDAT3 ;INDICATE EXPECTED DATA
3793 024424 104413 ERROR ;ERROR
;-----
024426 024424 .WORD -2 ;WRITE CONTROL LOGIC TESTS
3794

```

3795
3796
3797
3798
3799
3800 024430 050522
3801 024432 050474
3802 024434 000000
3803 024436 000240
024440 005237 001472

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

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

3808

SBTTL TEST # 143 - WRITE CONTROL LOGIC INHIBIT TEST

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

024444
 024444 000004

024446 024456
 024450 060000
 024452 000000
 024454 060042
 024456 012737
 024464 004437
 024470 024640

001015 177746
 002424

TST143:

```

SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        .WORD 405 ;TEST START LOCATION
        .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
        .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
        .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
  
```

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

```

3809 024472 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS CACHE TAG FIELD BITS TO BE
3810 ;WRITTEN TO CHR<15:07> ONLY DURING
3811 ;THE DESTINATION MEMORY ACCESS
3812 ;OF AN INSTRUCTION
3813 024500 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE CACHE STORE
3814 024506 005737 057744 TST 57744
3815 024512 005737 077744 TST 77744 ;READ UPDATE;LOAD BIT PATTERN
3816 ;000000011 INTO TAG STORE LOCATION
3817 ;7762
3818 024516 005737 177744 TST 177744 ;ACCESS I/O PAGE BY READING CCR REGISTER.
3819 ;THE CACHE COULD DO AN UPDATE TO
3820 ;TAG STORE LOCATION 7762 BUT THE ACCESS
3821 ;TO I/O PAGE WILL INHIBIT WRITE CONTROL
3822 ;LOGIC
3823 024522 005737 057744 TST 57744 ;WRITE TAG STORE DATA FROM LOCATION
3824 ;7762 INTO CHR<15:07>.
3825 024526 013737 177752 050474 MOV CHR,CHR157 ;SAVE CHR DATA
3826 024534 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
        024536 000240 NOP ;FOR LOOP ON ERROR
3827 024540 105037 177750 CLRB CMR ;DISABLE MAINTENANCE MODE
3828 024544 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3829 024552 042737 000177 050474 BIC #177,CHR157 ;PREPARE CHR157 FOR ERROR CHECK
3830 024560 02737 000600 050474 CMP #600,CHR157 ;BITS 15:07 SHOULD BE BIT PATTERN 000000011
3831 024566 001423 BEQ 10$ ;PASS
3832 024570 012737 000007 002062 MOV #7,LOOP ;ERROR;PREPARE CHR157 FOR TYPEOUT
3833 024576 006237 050474 2$: ASR CHR157
3834 024602 042737 100000 050474 BIC #100000,CHR157
3835 024610 005337 002062 DEC LOOP
3836 024614 001370 BNE 2$
3837 024616 012737 000003 050522 MOV #3,EXDAT ;INDICATE EXPECTED DATA
3838 024624 104413 ERROR ;ERROR
        ;-----
        024626 024624 .WORD -2 ;WRITE CONTROL LOGIC TESTS
3839 ;READING TAGD<21:13> THRU CHR<15:07>
3840 ;DID NOT RESULT IN BIT PATTERN 000000011.
3841 ;PRINT CHR 15:07 EXPECTED
3842 024630 050522 EXDAT3
  
```

3843 024632 050474
3844 024634 000000
3845 024636 000240
024640 005237 001472

108:

CHR157
.WORD 0
NOP
INC \$TESTN

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

3859

```

.SBTTL TEST # 144 - WRITE CONTROL AND VALID STORE LOGIC TEST
*****
*TEST 144 WRITE CONTROL AND VALID STORE LOGIC TEST
* THIS TEST VERIFIES THE AREA OF WRITE CONTROL LOGIC AND VALID
* STORE LOGIC THAT IS CONCERNED WITH BYPASS OPERATIONS. A WIRE STRAP
* IS USED TO ALLOW OR INHIBIT INVALIDATION OF VALID STORE DURING
* READ BYPASS CONDITIONS. UNLESS SWITCH REGISTER 08 IS IMPLEMENTED,
* THIS TEST ASSUMES THAT STRAP W1 IS IN PLACE.
*PROCEDURE: IF SWR 08 IS NOT IMPLEMENTED ,W1 IS ASSUMED IN PLACE. NO
* INVALIDATION OF VALID STORE SET A SHOULD OCCUR DUE TO READ MISS/BYPASS
* AND READ HIT/BYPASS CONDITIONS. 2. IF SWR 08 IS IMPLEMENTED, STRAP
* W2 IS ASSUMED IN PLACE. NO INVALIDATION SHOULD OCCUR DUE TO READ
* MISS/BYPASS, BUT INVALIDATION SHOULD OCCUR DUE TO READ HIT/BYPASS
* CONDITION.
*****

```

```

024644
024644 000004
024646 024656
024650 070026
024652 000000
024654 070112
024656 012737 001015 177746
024664 004437 002452
024670 025112

```

```

TST144:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

3860 024672 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
3861 024700 001407 BEQ 1$ ;YES
3862 024702 052737 000400 177746 BIS #FC,CCR ;NO,FLUSH CACHE FOR SET A
3863 024710 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
3864 024716 001374 BNE 200$
3865 024720 012737 000015 177746 1$: MOV #15,CCR ;NO UCB SO AS TO ENABLE CACHE STORES
3866 024726 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS READ UPDATES,AND VALID
3867 ;STORE DATA TO BE WRITTEN TO CMR <12>
3868 ;ONLY DURING THE DESTINATION ACCESS
3869 ;OF AN INSTRUCTION.
3870 024734 005737 040000 TST 40000
3871 024740 005737 060000 TST 60000 ;READ UPDATE TO CACHE LOCATION 0000.
3872 ;WRITE 1 INTO VALID STORE LOCATION 0000.
3873 024744 052737 001000 177746 BIS #UCB,CCR ;BYPASS MODE
3874 024752 005737 040000 TST 40000 ;READ MISS/BYPASS;
3875 024756 005737 060000 TST 60000 ;LOAD VALID STORE LOCATION 0000 DATA
3876 ;RESULTING FROM PREVIOUS READ MISS/BYPASS
3877 ;INTO CMR<12>.
3878 ;THIS IS ALSO A READ HIT/BYPASS CONDITION.
3879 024762 013700 177750 MOV CMR,R0 ;SAVE CMR CONTENTS
3880 024766 005737 040000 TST 40000 ;LOAD VALID STORE LOCATION 0000
3881 ;DATA RESULTING FROM PREVIOUS READ HIT
3882 ;/BYPASS INTO CMR<12>.
3883 024772 013701 177750 MOV CMR,R1 ;SAVE CMR CONTENTS
3884 024776 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
3885 025004 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
025006 000240 NOP ;FOR LOOP ON ERROR
3886 025010 105037 177750 CLRB CMR ;DISABLE MAINTENANCE MODE

```


3924

..SBTTL TEST # 145 - WRITE HIT IN BYPASS MODE INVALIDATES CACHE LOCATION
:*****
:TEST 145 WRITE HIT IN BYPASS MODE INVALIDATES CACHE LOCATION
:*****
TEST145:

025116
025116 000004

025120 025130
025122 070026
025124 000000
025126 070102
025130 012737 001015 177746
025136 004437 002452
025142 025274

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

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

3925 025144 032737 020000 177746
3926 025152 001407
3927 025154 052737 000400 177746
3928 025162 032737 010C00 177746
3929 025170 001574
3930 025172 012700 000002
3931 025176 005001
3932 025200 012737 000015 177746
3933 025206 112737 000002 177750
3934
3935
3936
3937 025214 005737 040000
3938 025220 005711
3939
3940 025222 052737 001000 177746
3941 025230 010011
3942 025232 005711
3943
3944 025234 013702 177750
3945 025240 012737 001015 177746
3946 025246 000240
025250 000240
3947 025252 105037 177750
3948 025256 032702 010000
3949 025262 001403
3950 025264 104413
025266 025264
3951
3952
3953
3954 025270 000000
3955 025272 000240
025274 005237 001472

BIT #VSIU,CCR ;IS SET A BEING USED?
BEQ 1\$;YES
BIS #FC,CCR ;NO,FLUSH CACHE FOR SET A
200\$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
BNE 200\$
1\$: MOV #2,R0 ;DATA TO R0
CLR R1 ;ADDRESS 0 TO R1
MOV #15,CCR ;NO UCB SO AS TO ENABLE CACHE STORES
MOVVB #HODO,CMR ;HODO ALLOWS READ UPDATES,AND VALID
;STORE DATA TO BE WRITTEN TO CMR <12>
;ONLY DURING THE DESTINATION ACCESS
;OF AN INSTRUCTION.
TST 40000
TST (R1) ;READ UPDATE TO CACHE LOCATION 0000.
;WRITE 1 INTO VALID STORE LOCATION 0000.
BIS #UCB,CCR ;BYPASS MODE
MOV R0,(R1) ;WRITE HIT BYPASS TO LOC. 0 SHOULD INVALIDATE
TST (R1) ;LOAD VALID STORE LOCATION 0000 DATA
;RESULTING FROM PREVIOUS WRITE HIT/BYPASS INTO CMR<12>.
MOV CMR,R2 ;SAVE CMR CONTENTS
MOV #OFF,CCR ;DISABLE CACHE
25\$: NOP ;INSTRUCTION 'JMP 1\$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINTENANCE MODE
BIT #VLD,R2 ;CHECK FOR 0
BEQ 10\$;PASS
3950 ERROR ;ERROR
;-----
;WORD .-2
;WRITE CONTROL LOGIC TESTS
;WRITE HIT /BYPASS DID NOT INVALIDATE
;CACHE VALID STORE LOCATION
10\$: .WORD 0
NOP ;END OF TEST
INC \$TESTN ;INCREMENT TEST COUNTER

VERIFY CACHE DATA STORE RAM MEMORY IC'S

```

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

```

025300
025300 000004

025302 025312
025304 070000
025306 000000
025310 070100
025312 012737
025320 004437
025324 025614

001015 177746
002452

```

TST146:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST

```

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

3971 025326 012700 060000
3972 025332 005020
3973 025334 020027 070000
3974 025340 001374
3975 025342 012700 060000
3976 025346 012701 040000
3977 025352 012702 177777
3978 025356 112737 000002 177750
3979
3980
3981
3982 025364 012737 000015 177746
3983 025372 005721
3984 025374 005720
3985 025376 020027 070000
3986 025402 001373
3987 025404 012700 060000
3988 025410 005710
3989
3990
3991 025412 013705 177754
3992 025416 010210
3993
3994 025420 005710
3995
3996 025422 013703 177754
3997 025426 000240

```

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
MOV#B #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
6$: TST (R1)+ ;UPDATE ALL LOW CACHE DATA STORE WITH 0'S
TST (R0)+ ;
CMP R0,#70000 ;DONE?
BNE 6$ ;NO
MOV #60000,R0 ;ADDR. 60000 TO R0
7$: TST (R0) ;READ HIT TO CACHE DATA STORE LOCATION
;SPECIFIED BY R0.CLOCK DATA STORE
;BITS INTO CDR<15:0>.SHOULD BE ALL 0'S.
MOV CDR,R5 ;SAVE CDR CONTENTS
MOV R2,(R0) ;WRITE HIT CACUSES UPDATE TO CACHE DATA
;STORE LOCATION.WRITE ALL 1'S.
TST (R0) ;READ HIT.CLOCK DATA STORE BITS TO
;CDR <15:0>.SHOULD BE ALL 1'S.
MOV CDR,R3 ;SAVE CDR CONTENTS
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE

```

3998	025430	000240				NOP			:FOR LOOP ON ERROR
3999	025432	005705				TST	R5		:SHOULD BE ALL 0'S
4000	025434	001424				BEQ	8\$:PASS
4001	025436	012737	001015	177746		MOV	#OFF,CCR		:DISABLE CACHE
4002	025444	105037	177750			CLRB	CMR		:CLEAR MAINT. MODE
4003	025450	005037	050504			CLR	EXDAT6		:SPECIFY EXPECTED CACHE DATA STORE DATA
4004	025454	010537	050502			MOV	R5,CDR150		:SPECIFY CACHE DATA STORE DATA READ THRU (DR<15:0>
4005	025460	010037	050506			MOV	R0,CA121		:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4006	025464	006237	050506			ASR	CA121		
	025470	104413				ERROR			:ERROR
									:-----
4007	025472	025470				.WORD	.-2		:DATA STORE MARCH PATTERN TEST
4008									:READING CACHE DATA STORE DATA
4009									:THRU CDR<15:0> DID NOT READ ALL 0'S.
4010									:THIS SUGGESTS THAT A RAM LOCATION
4011									:SPECIFIED BY CA121 WAS OVERWRITTEN
4012									:WITH A 1 WHEN WRITING A 1 TO ANOTHER
4013									:LOCATION.ANY BIT IN CDR150 DATA
4014									:THAT IS A 1 MAY POINT TO A BAD
4015									:CACHE DATA STORE RAM.
4016	025474	050504				EXDAT6			: EXPECTED CACHE DATA STORE DATA
4017	025476	050502				CDR150			: CACHE DATA STORE DATA READ
4018									:THRU CDR<15:0>
4019	025500	050506				CA121			:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4020	025502	000000				.WORD	0		
4021	025504	000435				BR	3\$:END THE TEST
4022	025506	022703	177777		8\$:	CMR	#-1,R3		:SHOULD BE ALL 1'S
4023	025512	001425				BEQ	9\$:PASS
4024	025514	012737	001015	177746		MOV	#OFF,CCR		:DISABLE CACHE
4025	025522	105037	177750			CLRB	CMR		:CLEAR MAINT. MODE
4026	025526	012737	177777	050504		MOV	#-1,EXDAT6		:SPECIFY EXPECTED CACHE DATA STORE DATA
4027	025534	010337	050502			MOV	R3,CDR150		:SPECIFY CACHE DATA STORE DATA READ
4028									:THRU CDR<15:0>
4029	025540	010037	050506			MOV	R0,CA121		:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4030	025544	006237	050506			ASR	CA121		
4031	025550	104413				ERROR			:ERROR
									:-----
4032	025552	025550				.WORD	.-2		:DATA STORE MARCH PATTERN TEST
4033									:READING CACHE DATA STORE DATA
4034									:THRU CDR<15:0> DID NOT READ ALL 1'S.
4035									:ANY BIT IN CDR150 DATA
4036									:THAT IS A 0 MAY POINT TO A BAD
4037									:CACHE DATA STORE RAM.
4038									
4039	025554	050504				EXDAT6			: EXPECTED CACHE DATA STORE DATA
4040	025556	050502				CDR150			: CACHE DATA STORE DATA READ
4041									:THRU CDR<15:0>
4042	025560	050506				CA121			:SPECIFY FAILED DATA STORE ADDRESS LOCATION
4043	025562	000000				.WORD	0		
4044	025564	000405				BR	3\$:END TEST
4045	025566	062700	000002		9\$:	ADD	#2,R0		:NEXT LOCATION
4046	025572	022700	070000			CMR	#70000,R0		:HAS ALL LO CACHE BEEN DONE?
4047	025576	001304				BNE	7\$:NO,CONTINUE
4048	025600	012737	001015	177746	3\$:	MOV	#OFF,CCR		:DISABLE CACHE
4049	025606	105037	177750			CLRB	CMR		:DISABLE MAINT. MODE

4050 025612 00C240
025614 005237 001472

108: NOP
INC STESTN

:END OF TEST
:INCREMENT TEST COUNTER

```

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

```

```

025620
025620 000004 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
025622 025632 .WORD 40$ ;TEST START LOCATION
025624 060000 .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
025626 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
025630 060100 .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
025632 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
025640 004437 002424 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
025644 026134 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

4066 025646 012700 070000 1$: MOV #70000,R0 ;ADDRESS LOC. 70000 TO R0
4067 025652 005020 5$: CLR (R0)+ ;CLEAR ALL HIGH CACHE MAIN MEMORY
4068 025654 020027 100000 CMP R0,#100000 ;DONE?
4069 025660 001374 BNE 5$ ;NO
4070 025662 012700 070000 MOV #70000,R0 ;ADDR. LOC. 70000 TO R0
4071 025666 012701 050000 MOV #50000,R1 ;ADDR. LOC. 50000 TO R1
4072 025672 012702 177777 MOV #-1,R2 ;R2 CONTAINS ALL 1'S
4073 025676 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & DATA STORE BITS
4074 ;TO BE WRITTEN TO CDR<15:0> ONLY DURING
4075 ;THE DESTINATION MEMORY ACCESS OF AN
4076 ;INSTRUCTION
4077 025704 012737 000015 177746 MOV #15,CCR ;WRITE ENABLE CACHE DATA STORES
4078 025712 005721 6$: TST (R1)+ ;UPDATE ALL HIGH CACHE DATA STORE WITH 0'S
4079 025714 005720 TST (R0)+ ;
4080 025716 020027 100000 CMP R0,#100000 ;DONE?
4081 025722 001373 BNE 6$ ;NO
4082 025724 012700 070000 MOV #70000,R0 ;ADDR. 70000 TO R0
4083 025730 005710 7$: TST (R0) ;READ HIT TO CACHE DATA STORE LOCATION
4084 ;SPECIFIED BY R0.CLOCK DATA STORE
4085 ;BITS INTO CDR<15:0>.SHOULD BE ALL 0'S.
4086 025732 013705 177754 MOV CDR,R5 ;SAVE CDR CONTENTS
4087 025736 010210 MOV R2,(R0) ;WRITE HIT CACUSES UPDATE TO CACHE DATA
4088 ;STORE LOCATION.WRITE ALL 1'S.
4089 025740 005710 TST (R0) ;READ HIT.CHICK DATA STORE BITS TO
4090 ;CDR <15:0>.SHOULD BE ALL 1'S.
4091 025742 013703 177754 MOV CDR,R3 ;SAVE CDR CONTENTS
4092 025746 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE

```

```

4093 025750 000240      NOP      ;FOR LOOP ON ERROR
4094 025752 005705      TST      R5      ;SHOULD BE ALL 0'S
4095 025754 001424      BEQ      8$      ;PASS
4096 025756 012737 001015 177746  MOV      #OFF,CCR ;DISABLE CACHE
4097 025764 105037 177750  CLR      CMR      ;CLEAR MAINT. MODE
4098 025770 005037 050504  CLR      EXDAT6   ;SPECIFY EXPECTED CACHE DATA STORE DATA
4099 025774 010537 050502  MOV      R5,CDR150 ;SPECIFY CACHE DATA STORE DATA READ
4100 026000 010037 050506  MOV      R0,CA121 ;THRU CDR<15:0>
4101 026004 006237 050506  ASR      CA121    ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4102 026010 104413      ERROR     ;ERROR
                                ;-----
026012 026010      .WORD    .-2
4103                                ;DATA STORE MARCH PATTERN TEST
4104                                ;READING CACHE DATA STORE DATA
4105                                ;THRU CDR<15:0> DID NOT READ ALL 0'S.
4106                                ;THIS SUGGESTS THAT A RAM LOCATION
4107                                ;SPECIFIED BY CA121 WAS OVERWRITTEN
4108                                ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
4109                                ;LOCATION.ANY BIT IN CDR150 DATA
4110                                ;THAT IS A 1 MAY POINT TO A BAD
4111                                ;CACHE DATA STORE RAM.
4112
4113 026014 050504      EXDAT6   ; EXPECTED CACHE DATA STORE DATA
4114 026016 050502      CDR150  ; CACHE DATA STORE DATA READ
4115                                ;THRU CDR<15:0>
4116 026020 050506      CA121   ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4117 026022 000000      .WORD    0
4118 026024 000435      BR       3$
4119 026026 022703 177777 8$:    CMP      #-1,R3 ;END THE TEST
4120 026032 001425      BEQ      9$      ;SHOULD BE ALL 1'S
4121 026034 012737 001015 177746  MOV      #OFF,CCR ;PASS
4122 026042 105037 177750  CLR      CMR      ;DISABLE CACHE
4123 026046 012737 177777 050504  MOV      #-1,EXDAT6 ;CLEAR MAINT. MODE
4124 026054 010337 050502  MOV      R3,CDR150 ;SPECIFY EXPECTED CACHE DATA STORE DATA
4125                                ;SPECIFY CACHE DATA STORE DATA READ
4126 026060 010037 050506  MOV      R0,CA121 ;THRU CDR<15:0>
4127 026064 006237 050506  ASR      CA121    ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4128 026070 104413      ERROR     ;ERROR
                                ;-----
026072 026070      .WORD    .-2
4129                                ;DATA STORE MARCH PATTERN TEST
4130                                ;READING CACHE DATA STORE DATA
4131                                ;THRU CDR<15:0> DID NOT READ ALL 1'S.
4132                                ;ANY BIT IN CDR150 DATA
4133                                ;THAT IS A 0 MAY POINT TO A BAD
4134                                ;CACHE DATA STORE RAM.
4135
4136 026074 050504      EXDAT6   ; EXPECTED CACHE DATA STORE DATA
4137 026076 050502      CDR150  ; CACHE DATA STORE DATA READ
4138                                ;THRU CDR<15:0>
4139 026100 050506      CA121   ;SPECIFY FAILED DATA STORE ADDRESS LOCATION
4140 026102 000000      .WORD    0
4141 026104 000405      BR       3$
4142 026106 062700 000002 9$:    ADD      #2,R0    ;END TEST
4143 026112 022700 100000  CMP      #100000,R0 ;NEXT LOCATION
4144 026116 001304      BNE     7$      ;HAS ALL HI CACHE BEEN DONE?
                                ;NO,CONTINUE

```

4145	026120	012737	001015	177746	38:	MOV	#OFF,CCR	:DISABLE CACHE
4146	026126	105037	177750			CLRB	CMR	:DISABLE MAINT. MODE
4147	026132	000240			108:	NOP		:END OF TEST
	026134	005237	001472			INC	STESTN	:INCREMENT TEST COUNTER

462

```

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

```

```

026140
026140 000004          SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
026142 026152          .WORD 40$          ;TEST START LOCATION
026144 070000          .WORD 1$-40$+67764        ;LOOP ON ERROR START LOCATION
026146 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
026150 070070          .WORD 25$-40$+67764        ;LOOP ON ERROR END LOCATION
026152 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
026160 004437 002452 JSR R4,RELC TH ;LOCATE TEST CODE TO HIGH CACHE SPACE
026164 026530          .WORD 10$+2        ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

4163 026166 005037 177752 1$: CLR CHR ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
4164 026172 012700 060000 MOV #60000,R0 ;ADDR. LOC. 60000 TO R0
4165 026176 012701 040000 MOV #40000,R1 ;ADDR. LOC. 40000 TO R1
4166 026202 112737 000003 177750 MOVB #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & TAG STORE BITS
4167 ;TO BE WRITTEN TO CHR<15:7> ONLY DURING
4168 ;THE DESTINATION MEMORY ACCESS OF AN INSTRUCTION
4169 ;TDAR ALLOWS TAG FIELD TO BE WRITTEN INTO FROM AMR<8:0>
4170 026210 012737 000015 177746 MOV #15,CCR ;WRITE ENABLE CACHE TAG STORES
4171 026216 005721 6$: TST (R1)+ ;WRITE ALL LOW CACHE TAG STORE WITH 0'S
4172 026220 005720 TST (R0)+
4173 026222 020027 070000 CMP R0,#70000 ;DONE?
4174 026226 001373 BNE 6$ ;NO
4175 026230 012737 177777 177752 MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
4176 026236 012700 060000 MOV #60000,R0 ;ADDR. 60000 TO R0
4177 026242 005710 7$: TST (R0) ;READ MISS TO CACHE TAG STORE LOCATION
4178 ;SPECIFIED BY R0.CLOCK TAG STORE
4179 ;BITS INTO CHR<15:7>.SHOULD BE ALL 0'S.
4180 ;ALSO CAUSES TAG STORE LOCATION TO BE
4181 ;WRITTEN WITH 1'S FROM AMR<8:0>
4182 026244 013705 177752 MOV CHR,R5 ;SAVE CHR CONTENTS
4183 026250 005710 TST (R0) ;READ MISS.CLOCK TAG STORE BITS TO
4184 ;CHR <15:0>.SHOULD BE ALL 1'S.
4185 026252 013703 177752 MOV CHR,R3 ;SAVE CHR CONTENTS
4186 026256 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
4187 026262 042705 000177 BIC #177,R5 ;FOR LOOP ON ERROR
4188 026266 005705 TST R5 ;SHOULD BE ALL 0'S

```



```

4189 026270 001437          BEQ      8$          ;PASS
4190 026272 012737 001015 177746  MOV     #OFF,CCR     ;DISABLE CACHE
4191 026300 105037 177750          CLR8    CMR         ;CLEAR MAINT. MODE
4192 026304 005037 050522          CLR     EXDAT3      ;SPECIFY EXPECTED CACHE TAG STORE DATA
4193 026310 010537 050474          MOV     R5,CHR157   ;SPECIFY CACHE TAG STORE TAG READ
4194                                ;THRU CHR<15:7>
4195 026314 012737 000007 002062  MOV     #7,LOOP     ;PREPARE CHR157 FOR TYPEOUT
4196 026322 006237 050474          ASR     CHR157
4197 026326 042737 100000 050474 4$:    BIC     #100000,CHR157
4198 026334 005337 002062          DEC     LOOP
4199 026340 001370          BNE     4$
4200 026342 010037 050506          MOV     R0,CA121   ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4201 026346 006237 050506          ASR     CA121
4202 026352 104413          ERROR      ;ERROR
                                ;-----
                                .WORD    .-2
4203                                ;TAG STORE MARCH PATTERN TEST
4204                                ;READING CACHE TAG STORE DATA
4205                                ;THRU CHR<15:7> DID NOT READ ALL 0'S.
4206                                ;THIS SUGGESTS THAT A RAM LOCATION
4207                                ;SPECIFIED BY CA121 WAS OVERWRITTEN
4208                                ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
4209                                ;LOCATION.ANY BIT IN CHR157 DATA
4210                                ;THAT IS A 1 MAY POINT TO A BAD
4211                                ;CACHE TAG STORE RAM.
4212
4213 026356 050522          EXDAT3      ; EXPECTED CACHE TAG STORE DATA
4214 026360 050474          CHR157      ; CACHE TAG STORE DATA READ
4215                                ;THRU CHR<15:7>
4216 026362 050506          CA121      ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4217 026364 000000          .WORD    0
4218 026366 000452          BR        3$
4219 026370 042703 000177          8$:    BIC     #177,R3          ;END THE TEST
4220 026374 022703 177600          CMP     #177600,R3 ;PREPARE R3 FOR CHECK
4221 026400 001440          BEQ     9$          ;SHOULD BE ALL 1'S
4222 026402 012737 001015 177746  MOV     #OFF,CCR     ;PASS
4223 026410 105037 177750          CLR8    CMR         ;DISABLE CACHE
4224 026414 012737 177777 050522  MOV     #-1,EXDAT3  ;CLEAR MAINT. MODE
4225 026422 010337 050474          MOV     R3,CHR157  ;SPECIFY EXPECTED CACHE TAG STORE DATA
4226 026426 012737 000007 002062  MOV     #7,LOOP     ;SPECIFY CACHE TAG STORE DATA THRU CHR<15:7>
4227 026434 006237 050474          5$:    ASR     CHR157   ;PREPARE CHR157 FOR TYPEOUT
4228 026440 042737 100000 050474  BIC     #100000,CHR157
4229 026446 005337 002062          DEC     LOOP
4230 026452 001370          BNE     5$
4231 026454 010037 050506          MOV     R0,CA121   ;SPECIFY FAILED TAG STORE ADDRESS LOCATION
4232 026460 006237 050506          ASR     CA121
4233 026464 104413          ERROR      ;ERROR
                                ;-----
                                .WORD    .-2
4234                                ;TAG STORE MARCH PATTERN TEST
4235                                ;READING CACHE TAG STORE DATA
4236                                ;THRU CHR<15:7> DID NOT READ ALL 1'S.
4237                                ;ANY BIT IN CHR157 DATA
4238                                ;THAT IS A 0 MAY POINT TO A BAD
4239                                ;CACHE TAG STORE RAM.
4240
4241 026470 050522          EXDAT3      ; EXPECTED CACHE TAG STORE DATA
    
```

```

4 4c
4243
4244 026472 050474
4245 026474 050506
4246 026476 000000
4247 026500 000405
4248 026502 062700 000002 9S:
4249 026506 022700 070000
4250 026512 001253
4251 026514 012737 001015 177746 3S:
4252 026522 105037 177750
4253 026526 000240
      026530 005237 001472 10S:

```

```

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

```

```

: CACHE TAG STORE DATA READ
: THRU CHR<15:7>
: CACHE TAG STORE READ THRU CHR<15:7>
: SPECIFY FAILED TAG STORE ADDRESS LOCATION
: END TEST
: NEXT LOCATION
: HAS ALL LO CACHE BEEN DONE?
: NO CONTINUE
: DISABLE CACHE
: DISABLE MAINT. MODE
: END OF TEST
: INCREMENT TEST COUNTER

```

4268

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

026534
026534 000004

026536 026546
026540 060000
026542 000000
026544 060070
026546 012737 001015 177746
026554 004437 002424
026560 027124

```
TST151:
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
MOV #OFF,CCR
JSR R4,RELCTL
;WORD 10$+2
```

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

```
4269 026562 005037 177752 1$: CLR CHR ;LOAD AMR<8:0> BY WRITING ALL 0'S TO CHR<8:0>
4270 026566 012700 070000 MOV #70000,R0 ;ADDR. LOC. 70000 TO R0
4271 026572 012701 050000 MOV #50000,R1 ;ADDR. LOC. 50000 TO R1
4272 026576 112737 000003 177750 MOVB #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & TAG STORE BITS
4273 ;TO BE WRITTEN TO CHR<15:7> ONLY DURING
4274 ;THE DESTINATION MEMORY ACCESS OF AN INSTRUCTION
4275 ;TDAR ALLOWS TAG FIELD TO BE WRITTEN INTO FROM AMR<8:0>
4276 026604 012737 000015 177746 MOV #15,CCR ;WRITE ENABLE CACJE TAG STORES
4277 026612 005721 6$: TST (R1)+ ;WRITE ALL HIGH CACHE TAG STORE WITH 0'S
4278 026614 005720 TST (R0)+ ;
4279 026616 020027 100000 CMP R0,#100000 ;DONE?
4280 026622 001373 BNE 6$ ;NO
4281 026624 012737 177777 177752 MOV #-1,CHR ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
4282 026632 012700 070000 MOV #70000,R0 ;ADDR. 70000 TO R0
4283 026636 005710 7$: TST (R0) ;READ MISS TO CACHE TAG STORE LOCATION
4284 ;SPECIFIED BY R0.CLOCK TAG STORE
4285 ;BITS INTO CHR<15:7>.SHOULD BE ALL 0'S.
4286 ;ALSO CAUSES TAG STORE LOCATION TO BE
4287 ;WRITTEN WITH 1'S FROM AMR<8:0>
4288 026640 013705 177752 MOV CHR,R5 ;SAVE CHR CONTENTS
4289 026644 005710 TST (R0) ;READ MISS.CLOCK TAG STORE BITS TO
4290 ;CHR <15:7>.SHOULD BE ALL 1'S.
4291 026646 013703 177752 MOV CHR,R3 ;SAVE CHR CONTENTS
4292 026652 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
026654 000240 NOP ;FOR LOOP ON ERROR
4293 026656 042705 000177 BIC #177,R5
4294 026662 005705 TST R5 ;SHOULD BE ALL 0'S
```

```

4295 026664 001437 BEQ 8$ :PASS
4296 026666 012737 001015 177746 MOV #OFF,CCR :DISABLE CACHE
4297 026674 105037 177750 CLR CMR :CLEAR MAINT. MODE
4298 026700 005037 050522 CLR EXDAT3 :SPECIFY EXPECTED CACHE TAG STORE DATA
4299 026704 010537 050474 MOV R5,CHR157 :SPECIFY CACHE TAG STORE TAG READ
4300 :THRU CHR<15:7>
4301 026710 012737 000007 002062 MOV #7,LOOP :PREPARE CHR157 FOR TYPEOUT
4302 026716 006237 050474 4$: ASR CHR157
4303 026722 042737 100000 050474 BIC #100000,CHR157
4304 026730 005337 002062 DEC LOOP
4305 026734 001370 BNE 4$
4306 026736 010037 050506 MOV R0,CA121 :SPECIFY FAILED TAG STORE ADDRESS LOCATION
4307 026742 006237 050506 ASR CA121
4308 026746 104413 ERROR :ERROR
:-----
026750 026746 .WORD .-2
4309 :TAG STORE MARCH PATTERN TEST
4310 :READING CACHE TAG STORE DATA
4311 :THRU CHR<15:7> DID NOT READ ALL 0'S.
4312 :THIS SUGGESTS THAT A RAM LOCATION
4313 :SPECIFIED BY CA121 WAS OVERWRITTEN
4314 :WITH A 1 WHEN WRITING A 1 TO ANOTHER
4315 :LOCATION.ANY BIT IN CHR157 DATA
4316 :THAT IS A 1 MAY POINT TO A BAD
4317 :CACHE TAG STORE RAM.
4318
4319 026752 050522 EXDAT3 : EXPECTED CACHE TAG STORE DATA
4320 026754 050474 CHR157 : CACHE TAG STORE DATA READ
4321 :THRU CHR<15:7>
4322 026756 050506 CA121 :SPECIFY FAILED TAG STORE ADDRESS LOCATION
4323 026760 000000 .WORD 0
4324 026762 000452 BR 3$ :END THE TEST
4325 026764 042703 000177 8$: BIC #177,R3 :PREPARE R3 FOR CHECK
4326 026770 022703 177600 CMP #177600,R3 :SHOULD BE ALL 1'S
4327 026774 001440 BEQ 9$ :PASS
4328 026776 012737 001015 177746 MOV #OFF,CCR :DISABLE CACHE
4329 027004 105037 177750 CLR CMR :CLEAR MAINT. MODE
4330 027010 012737 177777 050522 MOV #-1,EXDAT3 :SPECIFY EXPECTED CACHE TAG STORE DATA
4331 027016 010337 050474 MOV R3,CHR157 :SPECIFY CACHE TAG STORE DATA READ THRU CHR<15:7>
4332 027022 012737 000007 002062 MOV #7,LOOP :PREPARE CHR157 FOR TYPEOUT
4333 027030 006237 050474 5$: ASR CHR157
4334 027034 042737 100000 050474 BIC #100000,CHR157
4335 027042 005337 002062 DEC LOOP
4336 027046 001370 BNE 5$
4337 027050 010037 050506 MOV R0,CA121 :SPECIFY FAILED TAG STORE ADDRESS LOCATION
4338 027054 006237 050506 ASR CA121
4339 027060 104413 ERROR :ERROR
:-----
027062 027060 .WORD .-2
4340 :TAG STORE MARCH PATTERN TEST
4341 :READING CACHE TAG STORE DATA
4342 :THRU CHR<15:7> DID NOT READ ALL 1'S.
4343 :ANY BIT IN CHR157 DATA
4344 :THAT IS A 0 MAY POINT TO A BAD
4345 :CACHE TAG STORE RAM.
4346
4347 027064 050522 EXDAT3 : EXPECTED CACHE TAG STORE DATA

```

```

4348
4349
4350 027066 050474          CHR157
4351 027070 050506          CA121
4352 027072 000000          .WORD 0
4353 027074 000405          BR 3$
4354 027076 062700 000002    9$: ADD #2,R0
4355 027102 022700 100000    CMP #100000,R0
4356 027106 001253          BNE 7$
4357 027110 012737 001015 177746 3$: MOV #0'F,CCR
4358 027116 105037 177750    CLR# CMR
4359 027122 000240          NOP
      027124 005237 001472    INC $TESTN

```

```

: 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 HI CACHE BEEN DONE?
: NO, CONTINUE
: DISABLE CACHE
: DISABLE MAINT. MODE
: END OF TEST
: INCREMENT TEST COUNTER

```

4775

```

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

```

```

027130
027130 000004

027132 027142
027134 070004
027136 000000
027140 070040
027142 012737 001015 177746 40$:
027150 004437 002452
027154 027266

```

```

TST152:
          SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          .WORD 40$              ;TEST START LOCATION
          .WORD 1$-40$+67764     ;LOOP ON ERROR START LOCATION
          .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD 25$-40$+67764    ;LOOP ON ERROR END LOCATION
          MOV #OFF,CCR           ;DISABLE CACHE
          JSR R4,RELCTH          ;LOCATE TEST CODE TO HIGH CACHE SPACE
          .WORD 10$+2            ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

4376 027156 005037 060000
4377 027162 112737 000002 177750 1$:
4378
4379
4380
4381 027170 012737 000015 177746
4382 027176 005737 040000
4383 027202 005737 060000
4384
4385 027206 005737 060000
4386
4387 027212 013701 177750
4388 027216 000240 000240 25$:
027220 000240
4389 027222 105037 177750
4390 027226 012737 001015 177746
4391 027234 032701 004000
4392 027240 001403
4393 027242 104413

027244 027242
4394
4395
4396
4397
4398 027246 000000

```

```

          CLR 60000              ;0'S TO MAIN MEMORY LOCATION
          MOVB #HODO,CMR         ;ALLOWS UPPER AND LOWER BYTE DATA
          ;PARITY STORE BITS TO BE WRITTEN TO
          ;CMR<11:10> ONLY DURING THE DESTINATION
          ;ACCESS OF AN INSTRU. ION.
          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
          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
          ;-----
          .WORD -2
          ;DATA PARITY GEN. & STORE TESTS
          ;READING CACHE MAINT. REGISTER
          ;BIT 11 FOR UPPER BYTE PARITY DATA DID
          ;NOT RESULT IN 0.
          .WORD 0

```

4399 027250 032701 002000
 4400 027254 001403
 4401 027256 104413

 027260 027256
 4402
 4403
 4404
 4405
 4406 027262 000000
 4407 027264 000240
 027266 005237 001472

9\$: BIT #LPS,R1
 BEQ 10\$
 ERROR

 .WORD .-2

 10\$: .WORD 0
 NOP
 INC \$*ESTN

:CHECK 0 FOR LOWER BYTE PARITY DATA
 :PASS
 :ERROR
 :-----
 :DAT. PARITY GEN. \$ STORE TESTS
 :READING CACHE MAINT. REGISTER
 :BIT 10 FOR LOWER BYTE PARITY DATA DID
 :NOT RESULT IN 0.

 :END OF TEST
 :INCREMENT TEST COUNTER

4423

```

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

```

```

027272
027272 000004

027274 027304
027276 070014
027300 000000
027302 070050
027304 012737 001015 177746
027312 004437 002452
027316 027460

```

```

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

```

```

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

```

```

4424 027320 012737 000001 050516
4425 027326 013737 050516 060000
4426 027334 112737 000002 177750
4427
4428
4429
4430 027342 012737 000015 177746
4431
4432 027350 005737 040000
4433 027354 005737 060000
4434
4435
4436 027360 005737 060000
4437
4438 027364 013701 177750
4439 027370 000240
        027372 000240
4440 027374 105037 177750
4441 027400 012737 001015 177746
4442 027406 032701 004000
4443 027412 001404
4444 027414 104413

```

```

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

```

```

4445 027416 027414
4446

```

```

        .WORD -2
;DATA PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER

```



```

4447
4448
4449 027420 050516          FLTPAT
4450
4451 027422 000000          .WORD 0
4452 027424 032701 002000 8$: BIT #LPB,R1
4453 027430 001004          BNE 9$
4454 027432 104413          ERROR
                                :CHECK 1 FOR LOWER BYTE PARITY DATA
                                :PASS
                                :ERROR
                                :-----

                                027434 027432          .WORD .-2
                                :DAT. PARITY GEN. $ STORE TESTS
                                :READING CACHE MAINT. REGISTER
                                :BIT 10 FOR LOWER BYTE PARITY DATA DID
                                :NOT RESULT IN 1.
                                :PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
                                :GEN. INPUT: WRTD<15:0>

4455
4456
4457
4458
4459 027436 050516          FLTPAT
4460
4461 027440 000000          .WORD 0
4462 027442 006337 050516 9$: ASL FLTPAT
4463 027446 032737 000400 050516 BIT #400,FLTPAT
4464 027454 001724          BEQ 2$
4465 027456 000240          NOP
                                :NEXT PATTREN
                                :HAS PATTERN 000200 BEEN DONE
                                :NO
                                :END OF TEST
                                :INCREMENT TEST COUNTER
                                027460 005237 001472 10$: INC $TESTN
  
```


TEST # 154 - CHK THAT HI BYTE DATA PARITY GEN WRITES A 1

```

4505 027612 050516          FLTPAT          ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
4506                          ;GENERATOR INPUTS: WRTD<15:0>
4507 027614 000000          .WORD          0
4508 027616 032701 002000  8$: BIT          #LPS,R1      ;CHECK 0 FOR LOWER BYTE PARITY DATA
4509 027622 001404          BEQ          9$      ;PASS
4510 027624 104413          ERROR
                          ;-----
                          .WORD          -2
4511                          ;DAT. PARITY GEN. $ STORE TESTS
4512                          ;READING CACHE MAINT. REGISTER
4513                          ;BIT 10 FOR LOWER BYTE PARITY DATA DID
4514                          ;NOT RESULT IN 0.
4515 027630 050516          FLTPAT          ;PRINT FLOATING 1 PATTERN USED FOR DATA PARITY
4516                          ;GEN. INPUTS: WRTD<15:0>
4517 027632 000000          .WORD          0
4518 027634 006337 050516  9$: ASL          FLTPAT      ;NEXT PATTERN
4519 027640 103327          BCC          2$      ;CONTINUE IF PATTERN 100000 NOT DONE
4520 027642 000240          VOP
                          INC          $TESTN      ;END OF TEST
                          ;INCREMENT TEST COUNTER
027644 005237 001472

```

4532

```

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

```

```

027650
027650 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
027652 027662          .WORD 40$          ;TEST START LOCATION
027654 070004          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
027656 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
027660 070040          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
027662 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
027670 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
027674 030024          .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

```

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

```

4533 027676 005037 060000          CLR 60000 ;0'S TO MAIN MEMORY LOCATION
4534 027702 112737 000002 177750 1$: MOVB #HODO,CMR ;ALLOWS UPPER AND LOWER BYTE DATA
4535 ;PARITY STORE BITS TO BE WRITTEN TO
4536 ;CMR<11:10> ONLY DURING THE DESTINATION
4537 ;ACCESS OF AN INSTRUCTION.
4538 027710 012737 000115 177746 MOV #15+WWPD,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY.
4539 ;ENABLE WRITE WRONG PARITY DATA
4540 027716 005737 040000          TST 40000 ;
4541 027722 005737 060000          TST 60000 ;PLACE ALL 0'S ON WRD<15:0> INPUTS
4542 ;SINCE WWPD IS INVOKED A 1 WILL BE
4543 ;WRITTEN INTO PARITY STORE LOCATION 0000.
4544 027726 005737 060000          TST 60000 ;WRITE UPPER AND LOWER DATA PARITY BITS FROM
4545 ;LOCAT. 0000 INTO CMR<11:10> RESPECTIVELY.
4546 027732 013701 177750          MOV CMR,R1 ;SAVE CMR DATA
4547 027736 000240          NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
027740 000240          NOP ;FOR LOOP ON ERROR
4548 027742 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
4549 027750 105037 177750          CLRB CMR ;DISABLE MAINT. MODE
4550 027754 052737 000400 177746 BIS #FC,CCR ;BEFORE LEAVING TEST FLUSH CACHE TO
4551 ;REMOVE ANY EFFECTS OF WWPD
4552 027762 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT TILL DONE
4553 027770 001374          BNE 200$
4554 027772 032701 004000          BIT #HPB,R1 ;CHECK 1 FOR UPPER BYTE PARITY STORE.
4555 027776 001003          BNE 9$ ;PASS
4556 030000 104413          ERROR ;ERROR
                                ;-----
030002 030000          .WORD .-2
4557 ;DATA PARITY GEN. & STORE TESTS
4558 ;READING CACHE MAINT. REGISTER
4559 ;BIT 11 FOR UPPER BYTE PARITY DATA DID

```

4560
 4561 030004 000000
 4562 030006 032701 002000
 4563 030012 001003
 4564 030014 104413

9\$: .WORD 0
 BIT #LPB,R1
 BNE 10\$
 ERROR

;NOT RESULT IN 1.
 ;CHECK 1 FOR LOWER BYTE PARITY DATA
 ;PASS
 ;ERROR
 ;-----

030016 030014
 4565
 4566
 4567
 4568

.WORD -2

;DAT. PARITY GEN. \$ STORE TESTS
 ;READING CACHE MAINT. REGISTER
 ;BIT 10 FOR LOWER BYTE PARITY DATA DID
 ;NOT RESULT IN 1.

4569 030020 000000
 4570 030022 000240 001472
 030024 005237

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

;END OF TEST
 ;INCREMENT TEST COUNTER

4586

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

030030
 030030 000004

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

030032 030042
 030034 070004
 030036 000000
 030040 070040
 030042 012737 001015 177746
 030050 004437 002452
 030054 030152

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

4587 030056 005037 177752
 4588 030062 112737 000003 177750
 4589
 4590
 4591
 4592
 4593
 4594 030070 012737 000015 177746
 4595
 4596 030076 005737 040000
 4597 030102 005737 060000
 4598
 4599 030106 005737 060000
 4600 030112 013701 177750
 4601 030116 000240
 030120 000240
 4602 030122 105037 177750
 4603 030126 012737 001015 177746
 4604 030134 032701 001000
 4605 030140 001403
 4606 030142 104413

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

030144 030142
 4607
 4608
 4609

```
.WORD -2
;TAG PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 9 FOR TAG PARITY DATA DID
```

4610
4611 030146 000000
4612 030150 000240
030152 005237 001472

108: .WORD 0
NOP
INC \$TESTN

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

4529

```

.SBTTL TEST # 157 - VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1
*****
*TEST 157 VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1
* VERIFY THAT TAG PARITY GENERATOR WILL WRITE A 1 INTO TAG PARITY STORE
* ADDRESS 0000 FOR FLOATING 1 PATTERN ON TAG PARITY GENERATOR INPUTS
* PROCEDURE: GENERATE 1 FROM TAG PARITY
* GENERATOR BY PLACING FLOATING 1 PATTERN ON INPUTS
* AND WRITING 1 INTO TAG PARITY STORE
* LOCATION 0000.READ TAG PARITY STORE BIT FROM
* CMR<9>

```

CONDITIONS:

```

INPUTS TO TAG PARITY GEN:
TAG WRTD<21:13> FLOATING 1 PATTERN
WVPD(1)- 0
TAG PARITY STORE ADDRESS:
CA<12:1>-0000

```

RESULT: CMR<9>- 1

TST157:

```

030156
030156 000004
030160 030170
030162 070014
030164 000000
030166 070050
030170 012737 001015 177746
030176 004437 002452
030202 030326

```

```

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

```

```

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

```

```

4630 030204 012737 000001 050516
4631 030212 013737 050516 177752 2$:
4632
4633 030220 112737 000003 177750 1$:
4634
4635
4636
4637
4638
4639 030226 012737 000015 177746
4640
4641 030234 005737 040000
4642 030240 005737 060000
4643
4644 030244 005737 060000
4645 030250 013701 177750
4646 030254 000240
4647 030256 000240
4648 030260 105037 177750
4649 030264 012737 001015 177746
4650 030272 032701 001000
4651 030276 001004
030302 030300

```

```

MOV #1,FLTPAT ;1ST FLOATING PATTERN
MOV FLTPAT,CHR ;LOAD AMR<8:0> BY WRITING FLOATING
;PATTERN TO CHR<8:0>.
MOV# #HODO+TDAR,CMR ; HODO ALLOWS TAG
;PARITY STORE BIT TO BE WRITTEN TO
;CMR<9> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;TDAR ALLOWS INPUTS TO TAG PARITY STORE
;GENERATOR TO BE LOADED FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
;
TST 40000 ;
TST 60000 ;PLACE FLOATING 1 PATTERN ON TAG WRTD<21:13> INPUTS
;THEREBY WRITING 1 INTO PARITY STORE LOCATION 0000.
;WRITE TAG PARITY BIT FROM LOCAT. 0000 INTO CMR<9>
MOV CMR,R1 ;SAVE CMR DATA
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
BIT #TPB,R1 ;CHECK FOR 1
BNE 9$ ;PASS
ERROR ;ERROR
;-----
.WORD -2

```


4652
4653
4654
4655

4656 030304 050516

4657

4658 030306 000000

4659 030310 006337 050516

4660 030314 032737 001000 050516

4661 030322 001733

4662 030324 000240

030326 005237 001472

FLTPAT

.WORD 0

ASL FLTPAT

BIT #1000,FLTPAT

BEQ 2\$

10\$: NOP

INC \$TESTN

:TAG PARITY GEN. & STORE TESTS
:READING CACHE MAINT. REGISTER
:BIT 9 FOR TAG PARITY DATA DID
:NOT RESULT IN 1.
:PRINT FLOATING i PATTERN USED ON
:TAG PARITY GEN. INPUTS: TAG WRTD<21:13>

:NEXT PATTERN
:HAS PATTERN 400 BEEN DONE
:NO CONTINUE
:END OF TEST
:INCREMENT TEST COUNTER

```

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

```

```

030332
030332 000004
030334 030344
030336 070004
030340 000000
030342 070040
030344 012737 0C1015 177746
030352 004437 002452
030356 030472

```

```

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

```

```

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

```

```

4674 030360 005037 177752
4675 030364 112737 000003 177750 1$:
4676
4677
4678
4679
4680
4681 030372 012737 002015 177746
4682
4683 030400 005737 040000
4684 030404 005737 060000
4685
4686
4687 030410 005737 060000
4688 030414 013701 177750
4689 030420 000240 25$:
030422 000240
4690 030424 012737 001015 177746
4691 030432 105037 177750
4692 030436 052737 000400 177746
4693
4694 030444 032737 010000 177746 200$:
4695 030452 001374
4696 030454 032701 001000
4697 030460 001003
4698 030462 104413
030464 030462
4699
4700
4701

```

```

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
;
;PLACE ALL 0'S ON TAG WRD<20:13> INPUTS
;SINCE WWPT IS INVOKED A 1 WILL BE
;WRITTEN INTO PARITY STORE LOCATION 0000.
;WRITE TAG BIT FROM LOCAT. 0000 INTO CMR<9>
;SAVE CMR DATA
;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
;DISABLE CACHE
;DISABLE MAINT. MODE
;BEFORE LEAVING TEST FLUSH CACHE
;TO REMOVE ANY EFFECTS OF WWPT
;WAIT TILL DONE
;CHECK 1 FOR TAG PARITY STORE.
;PASS
;ERROR
;-----
;TAG PARITY GEN. & STORE TESTS
;READING CACHE MAINT. REGISTER
;BIT 9 FOR TAG PARITY DATA DID

```

4702
4703 030466 000000
4704 030470 000240
030472 005237 001472

108: .WORD 0
NOP
INC \$TESTN

:NOT RESULT IN 1.
:END OF TEST
:INCREMENT TEST COUNTER

4718

```

.SBTTL TEST # 161 - CLEAR ALL LOW CACHE DATA & TAG PARITY STORES
*****
TEST 161 CLEAR ALL LOW CACHE DATA & TAG PARITY STORES
WRITE AND READ 0'S TO ALL LOW CACHE DATA PARITY AND TAG PARITY STORES
CONDITIONS:
INPUTS TO DATA PARITY GEN:
WRD<15:0> ALL 0'S
WVPD(1)- 0
INPUTS TO TAG PARITY GEN.:
TAG WRD<21:13> ALL 0'S
WVPT(1)-0
DATA PARITY/TAG PARITY STORE ADDRESS:
CA<12:1>=0000 TO 3777
RESULT: CMR<11.9> ALL 0
*****
    
```

```

030476
030476 000004

030500 030510
030502 070030
030504 000000
030506 070056
030510 012737 001015 177746
030516 004437 002452
030522 030722
    
```

```

TST161: SCPCND ;SCOPE CONDITIONS:GO SET JP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
.WORD 40$ ;LOOP ON ERROR START LOCATION
.WORD 1$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 0 ;LOOP ON ERROR END LOCATION
.WORD 25$-40$+67764 ;DISABLE CACHE
MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE
JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST
.WORD 10$+2
    
```

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

```

4719 030524 005037 177752
4720 030530 012705 060000
4721 030534 005025
4722 030536 020527 070000
4723 030542 001374
4724 030544 012705 060000
4725 030550 012703 040000
4726 030554 112737 000003 177750
4727
4728
4729
4730
4731
4732 030562 012737 000015 177746
4733 030570 005713
4734 030572 005715
4735
4736 030574 005715
4737 030576 013701 177750
4738 030602 000240
030604 000240
4739 030606 105037 177750
4740 030612 012737 001015 177746
4741 030620 010537 050506
4742 030624 006237 050506
4743 030630 032701 004000
4744 030634 001404
4745 030636 104413
    
```

```

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$: MOVB #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY
; STORE BITS TO BE WRITTEN TO
;CMR<11:9> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;TDAR ALLOWS TAG PARITY STORE GENERATOR
;INPUTS TO BE LOADED FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE PARITY STORE
MOV #15,CCR
TST (R3)
TST (R5)
;WRITE 0'S INTO DATA/TAG PARITY STORE
;ADDRESS LOCATION SPECIFIED BY R5
;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
25$: MOV CMR,R1 ;SAVE CMR DATA
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
MOV R5,CA121 ;GET PARITY ADDRESS LOCATION USED
ASR CA121 ;PREPARE CA121 FOR TYPEOUT
BIT #HPB,R1 ;CHECK 0 HI BYTE PARITY STORE
BEQ 7$ ;PASS
ERROR ;ERROR
    
```

```

030640 030636          .WORD  .-2          :-----
4746                                     ;DATA/TAG PARITY GEN. & STORE TESTS
4747                                     ;READING CACHE MAINT. REGISTER
4748                                     ;BIT 11 FOR UPPER BYTE PARITY DATA DID
4749                                     ;NOT RESULT IN 0.
4750 030642 050506      CA121              ;PRINT PARITY STORE ADDRESS LOCATION
4751                                     ;USED: CA<12:1>
4752 030644 000000          .WORD  0          ;CHECK 0 FOR LOWER BYTE PARITY DATA
4753 030646 032701 002000  7$:  BIT      #LPB,R1  ;PASS
4754 030652 001404          BEQ      8$          ;ERROR
4755 030654 104413          ERROR          ;-----

030656 030654          .WORD  .-2          ;DATA/TAG PARITY GEN. $ STORE TESTS
4756                                     ;READING CACHE MAINT. REGISTER
4757                                     ;BIT 10 FOR LOWER BYTE PARITY DATA DID
4758                                     ;NOT RESULT IN 0.
4759                                     ;PRINT PARITY STORE ADDRESS USED:CA<12:1>
4760 030660 050506      CA121              ;CHECK 0 FOR TAG PARITY DATA
4761 030662 000000          .WORD  0          ;PASS
4762 030664 032701 001000  8$:  BIT      #TPB,R1  ;ERROR
4763 030670 001404          BEQ      9$          ;-----
4764 030672 104413          ERROR          ;DATA/TAG PARITY GEN. AND STORAGE TESTS
4765                                     ;READING CACHE MAINT.REGISTER BIT 9 FOR
4766                                     ;TAG PARITY DATA DID NOT RESULT IN 0.
4767                                     ;PRINT PARITY STORE ADDRESS USED: CA<12:1>
4768 030676 050506      CA121              ;NEXT PARITY STORE ADDRESS LOCATION
4769 030700 000000          .WORD  0          ;HAVE ALL LOW CACHE PARITY STORE ADDRESS
4770 030702 062705 000002  9$:  ADD      #2,R5  ;LOCATIONS BEEN DONE
4771 030706 062703 000002  ACD     #2,R3  ;NO,CONTINUE
4772 030712 022705 070000  CMP     #70000,R5 ;END OF TEST
4773                                     ;INCREMENT TEST COUNTER
4774 030716 001316          BNE     1$          ;
4775 030720 000240          NOP          ;
030722 005237 001472          INC     $TESTN  ;

```

4789

```

.SBTTL TEST # 162 - CLEAR ALL CACHE DATA & TAG PARITY STORES
*****
*TEST 162 CLEAR ALL CACHE DATA & TAG PARITY STORES
* WRITE AND READ 0'S TO ALL HI CACHE DATA PARITY AND TAG PARITY STORES
* CONDITIONS:
* INPUTS TO DATA PARITY GEN:
* WRD<15:0> ALL 0'S
* 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>=4000 TO 7777
* RESULT: CMR<11:9> ALL 0
*****

```

```

030726
030726 000004
030730 030740
030732 060030
030734 000000
030736 060056
030740 012737 001015 177746 40$:
030746 004437 002424
030752 031152
SCPCND
:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
:LOOP ON ERROR END LOCATION
:DISABLE CACHE
:LOCATE TEST CODE TO LOW CACHE SPACE
:ADDRESS OF START OF NEXT TEST

```

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

```

4790 030754 005037 177752
4791 030760 012705 070000
4792 030764 005025
4793 030766 020527 100000
4794 030772 001374
4795 030774 012705 070000
4796 031000 012703 050000
4797 031004 112737 000003 177750 1$:
4798
4799
4800
4801
4802
4803 031012 012737 000015 177746
4804
4805 031020 005713
4806 031022 005715
4807
4808 031024 005715
4809 031026 013701 177750
4810 031032 000240
4811 031036 105037 177750
4812 031042 012737 001015 177746
4813 031050 010537 050506
4814 031054 006237 050506
4815 031060 032701 004000
4816 031064 001404
CLR CHR ;LOAD AMR<8:0> WITH ALL 0'S
MOV #70000,R5 ;ADDRESS 70000 TO R5
2$: CLR (R5)+ ;CLEAR ALL HI CACHE MAIN MEMORY
CMP R5,#100000
BNE 2$
MOV #70000,R5 ;1ST ADDRESS LOCATION IN R5
MOV #50000,R3 ;ADDRESS 50000 IN R3
1$: MOVB #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY
; STORE BITS TO BE WRITTEN TO
;CMR<11:9> ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;TDAR ALLOWS TAG PARITY STORE GENERATOR
;INPUTS TO BE LOADED FROM AMR<8:0>
;NO UCB SO AS TO WRITE ENABLE PARITY
;STORE
;
;WRITE 0'S INTO DATA/TAG PARITY STORE
;ADDRESS LOCATION SPECIFIED BY R5
;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
25$: MOV CMR,R1 ;SAVE CMR DATA
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
CLRB CMR ;DISABLE MAINT. MODE
MOV #OFF,CCR ;DISABLE CACHE
MOV R5,CA121 ;GET PARITY ADDRESS LOCATION USED
ASR CA121 ;PREPARE CA121 FOR TYPEOUT
BIT #HPB,R1 ;CHECK 0 HI BYTE PARITY STORE
BEQ 7$ ;PASS

```


4861

```

.SBTTL TEST # 163 - CHK SETTING HI CACHE DATA & TAG PARITY STORES
*****
*TEST 163      CHK SETTING HI CACHE DATA & TAG PARITY STORES
*      WRITE AND READ 1'S TO ALL LOW CACHE DATA PARITY AND TAG PARITY STORES
*      CONDITIONS:
*          INPUTS TO DATA PARITY GEN:
*              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
*****
    
```

```

031156      031156      000004
031160      031170      .WORD      40$
031162      070034      .WORD      1$-40$+67764
031164      000000      .WORD      0
031166      070062      .WORD      25$-40$+67764
031170      012737      001015  177746  40$:  MOV      #OFF,CCR
031176      004437      002452      JSR      R4,RELCTH
031202      031406      .WORD      10$+2

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

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

4862 031204 012737 000001 177752      MOV      #1,CHR ;LOAD AMR<8:0> WITH BIT PATTERN 00000001
4863 031212 012705 060000      MOV      #60000,R5 ;ADDRESS 60000 TO R5
4864 031216 012725 000401      2$:  MOV      #401,(R5)+ ;WRITE A 401 IN ALL LOW CACHE MAIN MEMORY
4865 031222 020527 070000      CMP      R5,#70000
4866 031226 001373      BNE      2$
4867 031230 012705 060000      MOV      #60000,R5 ;1ST ADDRESS LOCATION IN R5
4868 031234 012703 040000      MOV      #40000,R3 ;ADDRESS 40000 IN R3
4869 031240 112737 000003 177750 1$:  MOVB     #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY
4870      ; STORE BITS TO BE WRITTEN TO
4871      ;CMR<11:9> ONLY DURING THE DESTINATION
4872      ;ACCESS OF AN INSTRUCTION.
4873      ;TDAR ALLOWS TAG PARITY STORE GENERATOR
4874      ;INPUTS TO BE LOADED FROM AMR<8:0>
4875 031246 012737 000015 177746      MOV      #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
4876 031254 005713      TST     (R3)
4877 031256 005715      TST     (R5)
4878      ;WRITE 1'S INTO DATA/TAG PARITY STORE
4879 031260 005715      TST     (R5) ;ADDRESS LOCATION SPECIFIED BY R5
4880 031262 013701 177750      MOV      CMR,R1 ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
4881 031266 000240      25$:  NOP      ;SAVE CMR DATA
031270 000240      NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
4882 031272 105037 177750      CLR     CMR ;FOR LOOP ON ERROR
4883 031276 012737 001015 177746      MOV      #OFF,CCR ;DISABLE MAINT. MODE
4884 031304 010537 050506      MOV      R5,CA121 ;DISABLE CACHE
4885 031310 006237 050506      ASR     CA121 ;GET PARITY ADDRESS LOCATION USED
4886 031314 032701 004000      BIT     #HPB,R1 ;PREPARE CA121 FOR TYPEOUT
4887 031320 001004      BNE     7$ ;CHECK 1 HI BYTE PARITY STORE
4888 031322 104413      ERROR   7$ ;PASS
;ERROR
    
```


4932

```

.SBTTL TEST # 164 - CHK SETTING HI CACHE DATA & TAG PARITY STORES
*****
*TEST 164      CHK SETTING HI CACHE DATA & TAG PARITY STORES
*   WRITE AND READ 1'S TO ALL HI CACHE DATA PARITY AND TAG PARITY STORES
*   CONDITIONS:
*       INPUTS TO DATA PARITY GEN:
*           WRTD<15:0>= 000401
*           WWPDP(1)= 0
*       INPUTS TO TAG PARITY GEN.:
*           TAG WRTD<21:13> =BIT PATTERN 000000001
*           WWPT(1)=0
*       DATA PARITY/TAG PARITY STORE ADDRESS:
*           CA<12:1>=4000 TO 7777
*   RESULT:    CMR<11:9> ALL 1'S
*****

```

```

031412
031412 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
031414 031424          .WORD 40$          ;TEST START LOCATION
031416 060034          .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
031420 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
031422 060062          .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
031424 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
031432 004437 002424   JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
031436 031642          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST

```

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

```

4933 031440 012737 000001 177752   MOV #1,CHR ;LOAD AMR<8:0> WITH BIT PATTERN 000000001
4934 031446 012705 070000           MOV #70000,R5 ;ADDRESS 70000 TO R5
4935 031452 012725 000401 2$: MOV #401,(R5)+ ;WRITE A 401 TO ALL HI CACHE
4936 031456 020527 100000           CMP R5,#100000
4937 031462 001373           BNE 2$
4938 031464 012705 070000           MOV #70000,R5 ;1ST ADDRESS LOCATION IN R5
4939 031470 012703 050000           MOV #50000,R3 ;ADDRESS 50000 IN R3
4940 031474 112737 000003 177750 1$: MOVB #HODO+TDAR,CMR ;HODO ALLOWS DATA PARITY/TAG PARITY
4941                                     ; STORE BITS TO BE WRITTEN TO
4942                                     ;CMR<11:9> ONLY DURING THE DESTINATION
4943                                     ;ACCESS OF AN INSTRUCTION.
4944                                     ;TDAR ALLOWS TAG PARITY STORE GENERATOR
4945                                     ;INPUTS TO BE LOADED FROM AMR<8:0>
4946 031502 012737 000015 177746   MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
4947 031510 005713           TST (R3)
4948 031512 005715           TST (R5)
4949                                     ;WRITE 1'S INTO DATA/TAG PARITY STORE
4950 031514 005715           TST (R5) ;ADDRESS LOCATION SPECIFIED BY R5
4951 031516 013701 177750           MOV CMR,R1 ;WRITE DATA/TAG PARITY BITS INTO CMR<11:9>
4952 031522 000240 25$: NOP ;SAVE CMR DATA
031524 000240           NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
4953 031526 105037 177750           CLRB CMR ;FOR LOOP ON ERROR
4954 031532 012737 001015 177746   MOV #OFF,CCR ;DISABLE MAINT. MODE
4955 031540 010537 050506           MOV R5,CA121 ;DISABLE CACHE
4956 031544 006237 050506           ASR CA121 ;GET PARITY ADDRESS LOCATION USED
4957 031550 032701 004000           BIT #HPB,R1 ;PREPARE CA121 FOR TYPEOUT
4958 031554 001004           BNE 7$ ;CHECK 1 HI BYTE PARITY STORE
4959 031556 104413           ERROR ;PASS
                                ;ERROR

```


5002

```

.SBTTL TEST # 165 - VERIFY BYTE DATA PARITY STORE ADDRESS LINES
*****
*TEST 165 VERIFY BYTE DATA PARITY STORE ADDRESS LINES
*VERIFY HI & LO BYTE DATA PARITY STORE ADDRESS LINES
*PROCEDURE: WRITE 0 INTO HI & LO BYTE DATA PARITY STORE
*ADDRESS LOCATION 0000.
*WRITE A 1 INTO HI & LO BYTE DATA PARITY STORE
*ADDRESS LOCAT. 0001.
*READ HI & LO BYTE DATA PARITY ADDRESS LOC.
*0000 FOR 0'S.
*REPEAT THE ABOVE SEQUENCE EACH TIME CHANGING THE
*ADDRESS LOCATION THE 1 IS WRITTEN INTO BY
*SHIFTING THE 1 ONE PLACE TO THE LEFT.
*****

```

```

031646
031646 000004

031650 031660
031652 070040
031654 000000
031656 070100
031660 012737
031666 004437
031672 032116

```

```

001015 177746
002452

```

```

TST165:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

5003 031674 000240
5004
5005
5006
5007 031676 005037 060000
5008 031702 012737 000002 050516
5009 031710 012702 040000
5010 031714 012703 060000
5011 031720 063702 050516
5012 031724 063703 050516
5013 031730 012713 000401
5014
5015 031734 112737 000002 177750
5016
5017
5018
5019 031742 012737 000015 177746
5020 031750 005737 040000
5021 031754 005737 060000
5022
5023 031760 005712
5024 031762 005713
5025
5026
5027 031764 005737 060000
5028
5029
5030 031770 013701 177750
5031 031774 000240

```

```

050516
2$:
1$:
25$:

```

```

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
MOV #40000,R2 ;ADDRESS 40000 INTO R2
MOV #60000,R3 ;ADDRESS 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
MOV# #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

TST (R2)
TST (R3) ;WRITE 1 INTO HI & LO BYTE DATA
;PARITY STORE LOCATION
;SPECIFIED BY R3'S BITS 1 THRU 12:CA<12:1>.
TST 60000 ;LOAD DATA FROM HI & LO BYTE PARITY
;DATA PARITY STORE LOCATION
;0000 INTO CMR<11:10>.
MOV CMR,R1 ;SAVE CMR DATA
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE

```

5032	031776	000240				NOP			:FOR LOOP ON ERROR
	032000	105037	177750			CLRB	CMR		:DISABLE MAINT. MODE
5033	032004	012737	001015	177746		MOV	#OFF,CCR		:DISABLE CACHE
5034	032012	032701	004000			BIT	#HPB,R1		:READING CMR<11> FOR HI BYTE
5035									:DATA PARITY STORE DATA SHOULD RESULT
5036									:IN 0.
5037	032016	001411				BEQ	8\$:PASS
5038	032020	013737	050516	050506		MOV	FLTPAT,CA121		:SAVE CA<12:1> USED
5039	032026	006237	050506			ASR	CA121		:PREPARE CA121 FOR TYPEOUT
5040	032032	104413				ERROR			:ERROR
									:-----
	032034	032032				.WORD	.-2		
5041									:HI & LO BYTE DATA PARITY STORE ADDRESS TEST
5042									:HI BYTE DATA PARITY STORE LOC. 0000
5043									:DID NOT READ AS A 0 INDICATING THAT
5044									:IT WAS OVERWRITTEN WITH A 1. THIS
5045									:SUGGESTS HI BYTE DATA PARITY STORE
5046									:ADDRESS LINE IS BAD.
5047	032036	050506				CA121			:PRINT PARITY STORE ADDRESS FAILURE
5048									:CA<12:1>.
5049									:NOTE THAT THE 1 IN THIS PATTERN
5050									:WILL POINT TO THE ADDRESS LINE OF
5051									:THAT BROUGHT OUT ERROR.
5052	032040	000000				.WORD	0		
5053	032042	032701	004000		8\$:	BIT	#HPB,R1		:READING CMR<10> FOR LO BYTE
5054									:DATA PARITY STORE DATA SHOULD RESULT
5055									:IN 0.
5056	032046	001411				BEQ	9\$:PASS
5057	032050	013737	050516	050506		MOV	FLTPAT,CA121		:SAVE CA<12:1> USED
5058	032056	006237	050506			ASR	CA121		:PREPARE CA121 FOR TYPEOUT
5059	032062	104413				ERROR			:ERROR
									:-----
	032064	032062				.WORD	.-2		
5060									:HI & LO BYTE DATA PARITY STORE ADDRESS TEST
5061									:LO BYTE DATA PARITY STORE LOC. 0000
5062									:DID NOT READ AS A 0 INDICATING THAT
5063									:IT WAS OVERWRITTEN WITH A 1. THIS
5064									:SUGGESTS HI BYTE DATA PARITY STORE
5065									:ADDRESS LINE IS BAD.
5066	032066	050506				CA121			:PRINT PARITY STORE ADDRESS FAILURE
5067									:CA<12:1>.
5068									:NOTE THAT THE 1 IN THIS PATTERN
5069									:WILL POINT TO THE ADDRESS LINE OF
5070									:THAT BROUGHT OUT ERROR.
5071	032070	000000				.WORD	0		
5072	032072	006337	050516		9\$:	ASL	FLTPAT		:NEXT PATTERN
5073	032076	022737	020000	050516		CMP	#20000,FLTPAT		:HAS DATA PARITY STORE ADDRESS 4000 BEEN DONE?
5074	032104	001301				BNE	2\$:NO
5075	032106	012737	000240	070000		MOV	#240,70000		:RESTORE OVERWRITTEN LOCATION 70000 WITH NOP.
5076	032114	000240			10\$:	NOP			:END OF TEST
	032116	005237	001472			INC	\$TESTN		:INCREMENT TEST COUNTER

5089

```

.SBTTL TEST # 166 - VERIFY TAG PARITY STORE ADDRESS LINES
*****
*TEST 166 VERIFY TAG PARITY STORE ADDRESS LINES
* VERIFY TAG PARITY STORE ADDRESS LINES
* PROCEDURE: WRITE 0 INTO TAG PARITY STORE
* ADDRESS LOCATION 0000.
* WRITE A 1 INTO TAG PARITY STORE
* ADDRESS LOCAT. 0001.
* READ TAG PARITY ADDRESS LOC.
* 0000 FOR 0'S.
* REPEAT THE ABOVE SEQUENCE ,EACH TIME CHANGING THE
* ADDRESS LOCATION THE 1 IS WRITTEN INTO BY
* SHIFTING THE 1 ONE PLACE TO THE LEFT.
*****

```

```

032122
032122 000004
032124 032134
032126 070026
032130 000000
032132 070066
032134 012737 001015 177746
032142 004437 002452
032146 032322

```

```

TST166:
          SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          .WORD 40$              ;TEST START LOCATION
          .WORD 1$-40$+67764    ;LOOP ON ERROR START LOCATION
          .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD 25$-40$+67764  ;LOOP ON ERROR END LOCATION
          MOV #OFF,CCR          ;DISABLE CACHE
          JSR R4,RELCTH        ;LOCATE TEST CODE TO HIGH CACHE SPACE
          .WORD 10$+2          ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

5090 032150 012737 000002 050516
5091 032156 012702 040000 2$:
5092 032162 012703 060000
5093 032166 063702 050516
5094 032172 063703 050516
5095 032176 112737 000002 177750 1$:
5096
5097
5098
5099 032204 012737 000015 177746
5100 032212 005737 040000
5101 032216 005737 060000
5102
5103 032222 005713
5104 032224 005712
5105
5106
5107 032226 005737 060000
5108
5109
5110 032232 013701 177750
5111 032236 000240 25$:
5112 032240 000240
5113 032242 105037 177750
5114 032246 012737 001015 177746
5115 032254 032701 001000
5116 032260 001411
5117 032262 013737 050516 050506

```

```

          MOV #2,FLTPAT        ;1ST FLOATING CACHE DATA STORE ADDRESS PATTERN
          MOV #40000,R2        ;ADDRESS 40000 INTO R2
          MOV #60000,R3        ;ADDRESS 60000 INTO R3
          ADD FLTPAT,R2        ;R2 CONTAINS 40000+FLTPAT
          ADD FLTPAT,R3        ;R3 CONTAINS 60000+FLTPAT
          MOVB #HODO,CMR      ;HODO ALLOWS TAG PARITY
          ;STORE BITS TO BE WRITTEN TO CMR<9>
          ; ONLY DURING THE
          ;DESTINATION MEMORY ACCESS OF AN INSTRUCTION.
          ;NO UCB SO AS TO WRITE ENABLE DATA PAPTITY STORE
          MOV #15,CCR
          TST 40000
          TST 60000          ;READ UPDATE: WRITE 0 INTO TAG
          ; PARITY STORE
          TST (R3)
          TST (R2)          ;WRITE 1 INTO TAG
          ;PARITY STORE LOCATION
          ;SPECIFIED BY R2'S BITS 1 THRU 12:CA<12:1>.
          TST 60000        ;LOAD DATA FROM
          ;TAG PARITY STORE LOCATION
          ;0000 INTO CMR<9>.
          MOV CMR,R1        ;SAVE CMR DATA
          NOP                ;INSTRUCTION 'JMP 1$' PLACED HERE
          NOP                ;FOR LOOP ON ERROR
          CLRB CMR          ;DISABLE MAINT. MODE
          MOV #OFF,CCR      ;DISABLE CACHE
          BIT #TFB,R1       ;READING CMR<9> FOR TAG PARITY STORE
          ;DATA SHOULD RESULT IN 0
          BEQ 9$            ;PASS
          MOV FLTPAT,CA121  ;SAVE CA<12:1> USED

```

```

5118 032270 006237 050506          ASR      CA121          :PREPARE CA121 FOR TYPEOUT
5119 032274 104413                   ERROR          :ERROR
                                     :-----
      032276 032274                   .WORD      .-2
5120                                     :TAG PARITY STORE ADDRESS TEST
5121                                     :TAG PARITY STORE LOC. 0000
5122                                     :DID NOT READ AS A 0 INDICATING THAT
5123                                     :IT WAS OVERWRITTEN WITH A 1. THIS
5124                                     :SUGGESTS TAG PARITY STORE
5125                                     :ADDRESS LINE IS BAD.
5126 032300 050506          CA121          :PRINT PARITY STORE ADDRESS FAILURE
5127                                     :CA<12:1>.
5128                                     :NOTE THAT THE 1 IN THIS PATTERN
5129                                     :WILL POINT TO THE ADDRESS LINE OF
5130                                     :THAT BROUGHT OUT ERROR.
5131 032302 000000                   .WORD      0
5132 032304 006337 050516          9$: ASL      FLTPAT          :NEXT PATTERN
5133 032310 022737 020000 050516 9$: CMP      #20000,FLTPAT :HAS TAG PARITY STORE ADDRESS 4000 BEEN DONE?
5134 032316 001317                   BNE      2$
5135 032320 000240          10$: NOP
      032322 005237 001472          INC      $TESTN      :INCREMENT TEST COUNTER

```

5140

```

.SBTTL TEST # 167 - PARITY ERROR BITS IN CME=0 AFTER WRITE TO CME
:*****
:TEST 167 PARITY ERROR BITS IN CME=0 AFTER WRITE TO CME
:* VERIFY THAT ALL PARITY ERROR BITS IN CACHE MEMORY ERROR REGISTER
:* WILL READ 0 FOLLOWING A WRITE TO CME.
:*****

```

```

032326
032326 000004

032330 032340
032332 070000
032334 000000
032336 070016
032340 012737 001015 177746 40$:
032346 004437 002452
032352 032424

```

```

TST167:
        SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
        .WORD 40$              ;TEST START LOCATION
        .WORD 1$-40$+67764     ;LOOP ON ERROR START LOCATION
        .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 25$-40$+67764    ;LOOP ON ERROR END LOCATION
        MOV #OFF,CCR           ;DISABLE CACHE
        JSR R4,RELCTH          ;LOCATE TEST CODE TO HIGH CACHE SPACE
        .WORD 10$+2           ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

5141 032354 112737 000002 177750 1$:  MOVB #HODO,CMR ;HODO WILL ALLOW CLOCKING OF PARITY INFO.
5142                                     ;TO CME ONLY DURING THE DESTINATION ACCESS OF AN
5143                                     ;INSTRUCTION. THE EFFECT
5144                                     ;IS THAT NO CLOCKING WILL OCCUR DURING EXECUTION
5145                                     ;OF THE NEXT INSTRUCTION.
5146 032362 005037 177744          CLR CME ;WRITE TO CME
5147 032366 013701 177744          MOV CME,R1 ;SAVE CME CONTENTS.
5148 032372 000240 25$:           NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP ;FOR LOOP ON ERROR
5149 032376 105037 177750          CLRB CMR ;DISABLE MAINTENANCE MODE
5150 032402 005701                TST R1 ;ARE ALL BITS 0?
5151 032404 001406                BEQ 10$ ;PASS
5152 032406 010137 050520          MOV R1,RECDAT ;GET CME CONTENTS RECEIVED
5153 032412 104413                ERROR ;ERROR
                                ;-----
                                .WORD -2
5154                                     ;PARITY ERROR CHECK TESTS
5155                                     ;WRITING TO CME DID NOT LEAVE ALL PARITY ERROR BITS 0
5156 032416 050520                RECDAT ;PRINT CME CONTENTS RECEIVED
5157 032420 000000                .WORD 0
5158 032422 000240 10$:           NOP ;END OF TEST
                                INC $TESTN ;INCREMENT TEST COUNTER
032424 005237 001472

```


5169

```

.SBTTL TEST # 170 - CME CAN SHOW NO PARITY ERROR FOLLOWING READ HIT
:*****
:*TEST 170 CME CAN SHOW NO PARITY ERROR FOLLOWING READ HIT
:* VERIFY THAT CME CAN SHOW NO PARITY ERRORS FOLOWING A READ HIT CONDITION
:* VERIFY TAG/DATA 'PARITY CHECK PARITY GENERATORS' WITH ALL 0'S
:* ON THEIR INPUTS.
:* PROCEDURE: CREATE ALL 0'S ON THE INPUTS OF THE TAG/DATA
:* PARITY CHECK PARITY GENERATORS DURING A READ
:* HIT CONDITION.ALLOW PARITY INFO. TO BE CLOKED 'O
:* CME.
:* RESULT: CME<15>,<7>,<6>,<5> ALL 0'S
:*****

```

032430
032430 000004

```

TST170:
        SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
        .WORD 40$              ;TEST START LOCATION
        .WORD 1$-40$+67764     ;LOOP ON ERROR START LOCATION
        .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 25$-40$+67764    ;LOOP ON ERROR END LOCATION
        MOV #OFF,CCR           ;DISABLE CACHE
        JSR R4,RELCTH          ;LOCATE TEST CODE TO HIGH CACHE SPACE
        .WORD 10$+2            ;ADDRESS OF START OF NEXT TEST

```

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

```

5170 032456 052737 000400 177746    BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5171 032464 032737 010000 177746    BIT #VCIP,CCR ;WAIT TILL DONE
5172 032472 001374                BNE 200$
5173 032474 005037 000000          CR 0 ;0'S TO MAIN MEMORY LOCATION 0
5174 032500 112737 000002 177750    MOVB #HODO,CMR ;HODO ALLOWS CLOKING OF PARITY INFO TO
5175                                ;CME ONLY DURING THE DESTINATION ACCESS OF
5176                                ;OF AN INSTRUCTION
5177 032506 012737 000015 177746    MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
5178 032514 005737 040000          TST 40000
5179 032520 005737 000000          TST 0
5180                                ;READ UPDATE TO CACHE LOCATION 0000-
5181                                ;ALL 0'S WILL BE WRITTEN INTO DATA/TAG STORES
5182 032524 005037 177744          CLR CME ;AND A 0 INTO DATA/TAG PARITY STORES
5183 032530 005737 000000          TST 0 ;CLEAR CME
5184                                ;READ HIT-ALL 0'S WILL BE PLACED ON INPUTS
5185                                ;OF DATA/TAG PARITY DETECT PARITY GENERATORS
5186 032534 052737 000200 177746    BIS #PEA,CCR ;AND ALL PARITY INFO WILL BE CLOKED TO CME
5187                                ;SET CCR<7> SO AS TO ENABLE CME<7>,<6>,<5> TO
5188                                ;TO BE WRITTEN INDIVIDUALLY FROM
5189                                ;PARITY INFO LOGIC,AND TO WRITE CME<15>.
5189 032542 000240                NOP
5190 032544 013701 177744          MOV CME,R1 ;SAVE CME CONTENTS
5191 032550 000240                NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
5192 032552 000240                NOP ;FOR LOOP ON ERROR
5192 032554 012737 001015 177746    MOV #OFF,CCR ;DISABLE CACHE
5193 032562 105037 177750          CLRB CMR ;DISABLE MAINTENANCE MODE
5194 032566 012737 000002 000000    MOV #2,0 ;RESTORE LOCATION 0
5195 032574 005701                TST R1 ;WERE ALL PARITY ERROR BITS IN CME 0?
5196 032576 001411                BEQ 10$ ;PASS;NEXT TEST
5197 032600 005037 050504          CLR EXDAT6 ;SPECIFY CME CONTENTS EXPECTED
5198 032604 010137 050520          MOV R1,RECDAT ;GET CME CONTENTS RECEIVED
5199 032610 104413                ERROR ;ERROR

```

5200 032612 032610
5201
5202 032614 050504
5203 032616 050520
5204 032620 000000
5205 032622 000240
032624 005237 001472

.WORD -2
EXDAT6
RECDAT
.WORD 0
10\$: NOP
INC \$TESTN

:-----
:PARITY CHECK TESTS
:ALL PARITY ERROR BITS IN CME SHOULD HAVE READ 0
:PRINT EXPECTED CME CONTENTS
:PRINT CONTENTS OF CME RECEIVED
:END OF TEST
:INCREMENT TEST COUNTER

```

.SBTTL TEST # 171 - PARITY ERROR BIT CHECK
*****
*TEST 171 PARITY ERROR BIT CHECK
* VERIFY THE FOLLOWING WHEN A LOCATION PREVIOUSLY WRITTEN
* WITH WRONG TAG PARITY IS ACCESSED:
* 1. A PARITY ERROR IS DETECTED AND ALL PARITY ERROR BITS IN
* CME READ CORRECTLY WITH PEA CLEARED.
* 2. ALL PARITY ERROR ERROR BITS READ CORRECTLY WITH PEA
* SET.
* 3. A WRITE TO CME CLEARS CME<15> AND <5> FROM A 1 STATE
PROCEDURE: WRITE WRONG PARITY TO TAG PARITY STORE LOCATION
0000. CLOCK PARITY INFO. TO CME.
CONDITIONS: DATA PARITY CHECK PARITY GEN. INPUTS:
ALL 0'S
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
*****

```

```

032630
032630 000004
032632 032642
032634 070016
032636 000000
032640 070126
032642 012737 001015 177746
032650 004437 002452
032654 033162

```

```

TST171:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
.WORD 40$ ;TEST START LOCATION
.WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
.WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
.WORD 10$+2 ;ADDRESS OF START OF NEXT TEST

```

```

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

```

```

5230 032656 052737 000400 177746 2$: BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5231 032664 032737 010000 177746 300$: BIT #VCIP,CCR ;WAIT TILL DONE
5232 032672 001374 BNE 300$
5233 032674 005037 000000 1$: CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
5234 032700 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CLOCKING OF PARITY INFO TO
;CME ONLY DURING THE DESTINATION ACCESS OF
5235 ;AN INSTRUCTION.
5236 ;NO UCB SO AS TO WRITE CACHE STORES
5237 032706 012737 000015 177746 MOV #15,CCR
5238 032714 005737 000000 TST 0
5239 032720 005737 040000 TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
5240 032724 052737 002000 177746 BIS #WWPT,CCR ;ALLOW WRITE WRONG PARITY TO TAGG PARITY STORE
5241 032732 005737 000000 TST 0 ;READ UPDATE TO CACHE LOCATION 0000;
5242 ;ALL 0'S(EVEN DATA) WILL BE WRITTEN TO DATA/TAG STORES,
5243 ;0'S INTO DATA PARITY STORES,AND A 1
5244 ;INTO TAG PARITY STORE.
5245 032736 042737 002000 177746 BIC #WWPT,CCR ;DISABLE WWPT
5246 032744 005037 177744 CLR CME ;CLEAR CME
5247 032750 005737 000000 TST 0 ;READ HIT; ALL 0'S WILL BE PLACED ON INPUTS

```


KKKAO 11-44 KK1'B CACHE
TEST # 171 - PARITY ERROR BIT CHECK

MACRO M1113 28-MAR-81 14:20 PAGE 143-2

K 16

SEQUENCE 205

5298
5299 033152 050504
5300 033154 050520
5301 033156 000000
5302 033160 000240
033162 005237 001472

105:

EXDAT6
RECDAT
.WORD 0
NOP
INC \$TESTN

:FOLLOWING WRITE TO CME
:PRINT CME CONTENTS EXPECTED
:PRINT CME CONTENTS RECEIVED
:
:END OF TEST
:INCREMENT TEST COUNTER

5329

```

.SBTTL TEST # 172 - VERIFY WRONG BYTE INFO CAUSES PROPER PARITY ERROR
*****
*TEST 172 VERIFY WRONG BYTE INFO CAUSES PROPER PARITY ERROR
*VERIFY THE FOLLOWING WHEN A LOCATION PREVIOUSLY WRITTEN
*WITH WRONG LO & HI BYTE PARITY IS ACCESSED.
* 1. A PARITY ERROR IS DETECTED AND ALL PARITY ERROR BITS IN
* CME READ CORRECTLY WITH PEA CLEARED.
* 2. ALL PARITY ERROR BITS READ CORRECTLY WITH PEA
* SET.
* 3. A WRITE TO CME CLEARS CME<15> ,<7> AND <6> FROM A 1 STATE
PROCEDURE: WRITE WRONG PARITY TO LO BYTE PARITY STORE LOCATION
          0000. CLOCK PARITY INFO. TO CME.
CONDITIONS: HI BYTE PARITY CHECK PARITY GEN. INPUTS:
            INTD<15:8>=0
            HI BYTE PARITY BIT=1
            LO BYTE PAR. CHECK PAR. GEN INPUTS:
            INTD<7:0>- ALL 0'S
            LO BYTE PARITY BIT=1
            TAG PARITY CHECK PARITY GEN. INPUTS:
            ALL 0'S
RESULTS:  PEA CLEARED:
          CME<15>-0
          CME<7>,<6>,<5>-1
          PEA SET:
          CME<15>=1
          CME<7>,<6>-1
          CME<5>=0
  
```

033166
 033166 000004
 033170 033200
 033172 070016
 033174 000000
 033176 070126
 033200 012737 001015 177746
 033206 004437 002452
 033212 033520

```

TST172:
        SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
        .WORD 40$              ;TEST START LOCATION
        .WORD 1$-40$+67764    ;LOOP ON ERROR START LOCATION
        .WORD 0                ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
        .WORD 25$-40$+67764   ;LOOP ON ERROR END LOCATION
        MOV #OFF,CCR          ;DISABLE CACHE
        JSR R4,RELCTH         ;LOCATE TEST CODE TO HIGH CACHE SPACE
        .WORD 10$ 2           ;ADDRESS OF START OF NEXT TEST
  
```

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

```

5330 033214 052737 000400 177746      BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5331 033222 032737 010000 177746      BIT #VCIP,CCR ;WAIT TILL DONE
5332 033230 001374                BNE 200$
5333 033232 005037 000000          CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
5334 033236 112737 000002 177750      MOV#B #HODO,CMR ;HODO ALLOWS CLOCKING OF PARITY INFO 10
5335                                ;CME ONLY DURING THE DESTINATION ACCESS OF
5336                                ;AN INSTRUCTION.
5337 033244 012737 000015 177746      MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
5338 033252 005737 000000          TST 0 ;
5339 033256 005737 040000          TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
5340 033262 052737 000100 177746      BIS #WWPD,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO
5341                                ;& HI BYTE PARITY STORE.
5342 033270 005737 000000          TST 0 ;READ UPDATE TO CACHE LOCATION 0000;
5343                                ;ALL 0'S WILL BE WRITTEN TO DATA/TAG STORES,
5344                                ;1'S INTO LO & HI BYTE DATA PARITY STORES,AND A 0
  
```

```

5345                                     :INTO TAG PARITY STORE.
5346 033274 042737 000100 177746      BIC  #WWPD,CCR      :DISABLE WWPD
5347 033302 005037 177744             CLR  CME           :CLEAR CME
5348 033306 005737 000000             TST  0            :READ HIT; ALL 0'S(EVEN DATA) WILL BE
5349                                     :PLACED ON INPUTS
5350                                     :OF TAG PARITY ERROR CHECK PARITY GEN'S, BUT
5351                                     :LO & HI BYTE PARITY CHECK GENERATORS WILL
5352                                     :SEE ODD DATA DUE TO WRONG PARITY
5353                                     :FROM PREVIOUS READ UPDATE.
5354 033312 013700 177744             MOV  CME,R0        :SAVE PARITY ERROR BITS WITH PEA CLEARED
5355 033316 052737 000200 177746      BIS  #PEA,CCR      :SET CCR<7> SO AS TO WRITE CME<7>,<6>,<5> INDIVIDUALLY
5356                                     :FROM PARITY CHECK LOGIC, AND TO WRITE CME<15>
5357 033324 000240                     NOP
5358 033326 013701 177744             MOV  CME,R1        :SAVE CME CONTENTS AFTER PEA IS SET.
5359                                     :LOOKS LIKE EXPECTED TRAP
5360                                     :DID NOT OCCUR.
5361 033332 005037 177744             CLR  CME           :CLEAR CME
5362 033336 013702 177744             MOV  CME,R2        :SAVE CME CONTENTS AFTER CLEAR
5363 033342 000240                     NOP                25$:
5364 033344 000240                     NOP                :INSTRUCTION 'JMP 1$' PLACED HERE
5365 033346 012737 001015 177746      MOV  #OFF,CCR      :FOR LOOP ON ERROR
5366 033354 105037 177750             CLR  CMR           :DISABLE CACHE
5367 033360 012737 000002 000000     MOV  #2,0          :DISABLE MAINT. MODE
5368 033366 052737 000400 177746      BIS  #FC,CCR       :RESTORE LOCATION 0
5369 033374 032737 010000 177746     BIT  #VCIP,CCR     :BEFORE LEAVING TEST FLUSH CACHE TO
5370 033402 001374                     BNE  4$            :ELIMINATE ANY EFFECTS OF WWPD
5371 033404 022700 000340             CMP  #340,R0       :WAIT TILL DONE
5372                                     :WERE PARITY ERROR BITS CORRECT IN CME
5373 033410 001412                     BEQ  7$            :WITH PEA CLEARED?
5374 033412 012737 000340 050504     MOV  #340,EXDAT6  :YES
5375 033420 010037 050520             MOV  R0,RECDAT    :SPECIFY CME CONTENTS EXPECTED
5376 033424 104413                     ERROR            :GET CME CONTENTS RECEIVED
5377                                     :ERROR
5378                                     :-----
5379                                     :
5380 033426 033424                     .WORD  -2
5381                                     :PARITY CHECK TESTS
5382                                     :PARITY ERROR BITS DID NOT READ CORRECTLY
5383                                     :WITH PEA CLEARED
5384                                     :PRINT CME CONTENTS EXPECTED
5385                                     :PRINT CME CONTENTS RECEIVED
5386 033430 050504                     EXDAT6
5387 033432 050520                     RECDAT
5388 033434 000000                     .WORD  0
5389 033436 022701 100300             CMP  #100300,R1   :WERE PARITY ERROR BITS CORRECT WITH PEA SET?
5390 033442 001412                     BEQ  8$            :YES
5391 033444 012737 100300 050504     MOV  #100300,EXDAT6 :SPECIFY CME CONTENTS EXPECTED
5392 033452 010137 050520             MOV  R1,RECDAT    :GET CME CONTENTS RECEIVED
5393 033456 104413                     ERROR            :ERROR
5394                                     :-----
5395                                     :
5396 033460 033456                     .WORD  -2
5397                                     :PARITY CHECK TESTS
5398                                     :PARITY ERROR BITS DID NOT READ CORRECTLY
5399                                     :WITH PEA SET.
5400                                     :PRINT CME CONTENTS EXPECTED
5401                                     :PRINT CME CONTENTS RECEIVED
5402 033462 050504                     EXDAT6
5403 033464 050520                     RECDAT
5404 033466 000000                     .WORD  0
5405 033470 005702                     TST  R2            :DID CME CLEAR?
5406 033472 001411                     BEQ  10$           :YES
5407 033474 005037 050504             CLR  EXDAT6       :SPECIFY CME CONTENTS EXPECTED
  
```

TRKACO 11-44 KK11B CACHE

MACRO M1113 28-MAR-81 14:20 PAGE 144-2

TEST # 172 - VERIFY WRONG BYTE INFO CAUSES PROPER PARITY ERROR

5397 033500 010237 050520
5398 033504 104413

MOV R2,RECDAT
ERROR

:GET CME CONTENTS RECEIVED
:ERROR
:-----

5399 033506 033504

.WORD -2

:PARITY CHECK TESTS
:PARITY ERROR BITS DID NOT CLEAR
:FOLLOWING WRITE TO CME
:PRINT CME CONTENTS EXPECTED
:PRINT CME CONTENTS RECEIVED

5400
5401

5402 033510 050504

EXDAT6
RECDAT

5403 033512 050520

.WORD 0

5404 033514 000000

5405 033516 000240

10\$:

NOP
INC \$TESTN

:END OF TEST
:INCREMENT TEST COUNTER

033520 005237 001472

5413

```
.SBTTL TEST # 173 - INT. LOGIC TRAPS ACCESSING LOC WITH BAD PARITY  
:*****  
*TEST 173 INT. LOGIC TRAPS ACCESSING LOC WITH BAD PARITY  
* VERIFY INTERRUPT LOGIC BY ASSURING THAT A TRAP OCCURS TO LOCATION  
* 114 WHEN A LOCATION PREVIOUSLY WRITTEN  
* WITH WRONG HI/LO BYTE PARITY IS ACCESSED.  
* CONDITIONS: PEA=0  
* DCPI 0  
:*****
```

033524
033524 000004

033526 033536
033530 070016
033532 000000
033534 070152
033536 012737 001015 177746
033544 004437 002452
033550 034010

```
TST173:  
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON  
;ERROR/LOOP ON TEST  
;TEST START LOCATION  
.WORD 40$ ;LOOP ON ERROR START LOCATION  
.WORD 1$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST  
.WORD 0 ;LOOP ON ERROR END LOCATION  
.WORD 25$-40$+67764 ;DISABLE CACHE  
MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE  
JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST  
.WORD 10$+2
```

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

5414	033552	052737	000400	177746	BIS	#FC,CCR	;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5415	033560	032737	010000	177746	200\$:	BIT	#VCIP,CCR
5416	033566	001374				BNE	200\$
5417	033570	005037	000000		1\$:	CLR	0 ;O'S TO MAIN MEMORY LOCATION 0.
5418	033574	012737	070142	000114		MOV	#4\$-40\$+67764,114 ;SETUP FOR CACHE TRAP
5419	033602	012737	000340	000116		MOV	#340,116
5420	033610	112737	000002	177750		MOVB	#HODO,CMR ;HODO ALLOWS CACHE UPDATES
5421							;AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC
5422							; ONLY DURING THE DESTINATION ACCESS OF AN INSTRUCTION.
5423	033616	005037	002070			CLR	FAIL1 ;CLEAR ERROR FLAG
5424	033622	012737	000015	177746		MOV	#15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
5425	033630	005737	000000			TST	0
5426	033634	005737	040000			TST	40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
5427	033640	052737	000100	177746		BIS	#WWPD,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO
5428							; & HI BYTE PARITY STORE.
5429	033646	005737	000000			TST	0 ;READ UPDATE TO CACHE LOCATION 0000;
5430							;WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES
5431	033652	042737	000100	177746		BIC	#WWPD,CCR ;DISABLE WWPD
5432	033660	005037	177744			CLR	CME ;CLEAR CME AND PARITY DETECT LOGIC
5433	033664	042737	000005	177746		BIC	#DCPI+FMLO,CCR ;ALLOW FOR INTERRUPT TO OCCUR
5434							;AND ENABLE LOW CACHE
5435	033672	005737	000000			TST	0 ;READ HIT;
5436							;LO & HI BYTE PARITY CHECK GENERATORS WILL
5437							; DETECT WRONG PARITY AND THE PARITY
5438							;ERROR WILL BE CLOCKED TO INTERRUPT
5439							;LOGIC
5440	033676	000240				NOP	
5441	033700	005237	002070			INC	FAIL1 ;INDICATE THAT TRAP DID NOT OCCUR
5442	033704	012737	001015	177746		MOV	#OFF,CCR ;DISABLE CACHE
5443	033712	000404				BR	25\$
5444	033714	012737	001015	177746	4\$:	MOV	#OFF,CCR ;DISABLE CACHE
5445	033722	022626				CMP	(SP)+,(SP)+ ;READJUST STACK DUE TO INTERRUPT
5446	033724	000240			25\$:	NOP	;INSTRUCTION 'JMP 1\$' PLACED HERE
	033726	000240				NOP	;FOR LOOP ON ERROR

5447	033730	105037	177750		CLRB	CMR		:DISABLE MAINT. MODE
5448	033734	012737	000002	000000	MOV	#2,0		:RESTORE LOCATION 0
5449	033742	012737	000116	000114	MOV	#116,114		:RESTORE CACHE INTERRUPT VECTORS
5450	033750	005037	000116		CLR	116		
5451	033754	052737	000400	177746	BIS	#FC,CCR		:BEFORE LEAVING TEST FLUSH CACHE TO
5452								:ELIMINATE ANY EFFECTS OF WWP
5453	033762	032737	010000	177746	500\$:	BIT	#VCIP,CCR	:WAIT TILL DONE
5454	033770	001374			BNE	500\$		
5455	033772	005737	002070		TST	FAIL1		:DID TRAP OCCUR?
5456	033776	001403			BEQ	10\$:YES
5457	034000	104413			ERROR			:ERROR
								:-----
	034002	034000			.WORD	.-2		
5458								: INTERRUPT/ABORT LOGIC TESTS
5459								: TRAP TO LOCATION 114 DID NOT OCCUR
5460	034004	000000			.WORD	0		
5461	034006	000240			NOP			:END OF TEST
	034010	005237	001472	10\$:	INC	\$TESTN		:INCREMENT TEST COUNTER

5469

```
.SBTTL TEST # 174 - VERIFY INTERRUPT LOGIC TRAP CAN BE INHIBITED
*****
*TEST 174 VERIFY INTERRUPT LOGIC TRAP CAN BE INHIBITED
* VERIFY INTERRUPT LOGIC BY ASSURING THAT A TRAP CAN BE INHIBITED TO LOCATION
* 114 WHEN A LOCATION PREVIOUSLY WRITTEN
* WITH WRONG HI/LO BYTE PARITY IS ACCESSED.
* CONDITIONS: PEA=0
* DCPI=1
*****
```

```
034014
034014 000004

034016 034026
034020 070016
034022 000000
034024 070152
034026 012737 001015 177746
034034 004437 002452
034040 034300
```

```
TST174:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
40$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
;WORD 10$+2
```

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

```
5470 034042 052737 000400 177746 BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5471 034050 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT TILL DONE
5472 034056 001374 BNE 200$
5473 034060 005037 000000 1$: CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
5474 034064 012737 070136 000114 MOV #4$-40$+67764,114 ;SETUP FOR CACHE TRAP
5475 034072 012737 000340 000116 MOV #340,116
5476 034100 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES
5477 ;AND CLOCKING OF PARITY INFO TO INTERRUPT LOGIC
5478 ; ONLY DURING THE DESTINATION ACCESS OF
5479 ;AN INSTRUCTION.
5480 034106 005037 002070 CLR FAIL1 ;CLEAR ERROR FLAG
5481 034112 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
5482 034120 005737 000000 TST 0
5483 034124 005737 040000 TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
5484 034130 052737 000100 177746 BIS #WWPD,CCR ;ALLOW WRITE WRONG PARITY DATA TO LO
5485 ;& HI BYTE PARITY STORE.
5486 034136 005737 000000 TST 0 ;READ UPDATE TO CACHE LOCATION 0000;
5487 ;WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES
5488 034142 042737 000100 177746 BIC #WWPD,CCR ;DISABLE WWPD
5489 034150 005037 177744 CLR CME ;CLEAR CME AND PARITY DETECT LOGIC
5490 034154 042737 000004 177746 BIC #FMLO,CCR ;ENABLE LO CACHE
5491 034162 005737 000000 TST 0 ;READ HIT;
5492 ;LO & HI BYTE PARITY CHECK GENERATORS WILL
5493 ; DETECT WRONG PARITY AND THE PARITY
5494 ;ERROR WILL BE CLOCKED TO INTERRUPT
5495 ;LOGIC
5496 034166 000240 NOP
5497 034170 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
5498 034176 000406 BR 25$
5499 034200 012737 001015 177746 4$: MOV #OFF,CCR ;DISABLE CACHE
5500 034206 005237 002070 INC FAIL1 ;INDICATE THAT TRAP OCCURED
5501 034212 022626 CMP (SP)+,(SP)+ ;ADJUST STACK DUE TO INTERRUPT
5502 034214 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
034216 000240 NOP ;FOR LOOP ON ERROR
```


53

```
.SBTTL TEST # 175 - CME BIT 15 OPERATES PROPERLY & TRAP TO 114 OCCURS
:*****
:*TEST 175 CME BIT 15 OPERATES PROPERLY & TRAP TO 114 OCCURS
:* VERIFY ABORT LOGIC BY THE FOLLOWING RESULTS WHEN A LOCATION
:* PREVIOUSLY WRITTEN WITH WRONG HI/LO BYTE PARITY IS ACCESSED.
:* 1. CME<15> WILL SET CAUSED BY ABORT SIGNAL BEING ASSERTED
:* 2. WRITE TO CME WILL CLEAR CME<15>
:* 3. INSTRUCTION CYCLE WILL BE ABORTED
:* 4. THE ABORT CAUSES TRAP TO 114
:* PROCEDURE: INHIBIT CLOCKING OF PARITY ERROR SIGNAL TO
:* INTERRUPT LOGIC. ALLOW CME<15> TO BE SET
:* BY ABORT SIGNAL WHICH IS ASSERTED BY PARITY
:* ERROR SIGNAL TO ABORT LOGIC.
:* CONDITIONS: PEA=1
:* DCFI=1
:*****
```

```
034304 000004
034306 034316
034310 070016
034312 000000
034314 070174
034316 012737 001015 177746 40$: MOV #OFF,CCR
034324 004437 002452 JSR R4,RELCTH
034330 034656 .WORD 10$+2
```

```
TST175:
: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

```
5534 034332 052737 000400 177746 BIS #FC,CCR ;FLUSH CACHE TO INVALIDATE ALL CACHE LOCATIONS
5535 034340 032737 010000 177746 200$: BIT #VCIP,CCR ;WAIT TILL DONE
5536 034346 001374 BNE 200$
5537 034350 005037 000000 1$: CLR 0 ;ALL 0'S TO LOCATION 0
5538 034354 005000 CLR R0 ;ADDRESS 0 TO R0
5539 034356 012737 070150 000114 MOV #4$-40$+67764,114 ;SETUP FOR TRAP
5540 034364 012737 000340 000116 MOV #340,116
5541 034372 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES
5542 ;AND CLOCK'NG OF PARITY INFO TO INTERRUPT LOGIC
5543 ; ONLY DURING THE DESTINATION ACCESS OF
5544 ;AN INSTRUCTION.
5545 034400 005037 002070 CLR FAJL1 ;CLEAR ERROR FLAG
5546 034404 012703 177777 MOV #-1,R3 ;ALL 1'S TO R3
5547 034410 012737 0000'5 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
5548 034416 005710 TST (R0)
5549 034420 005737 040000 TST 40000
5550 034424 052737 000100 177746 BIS #WVPD,CCR ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
5551 ;ALLOW WRITE WRONG PARITY DATA TO LO
5552 034432 005710 TST (R0) ;& HI BYTE PARITY STORE.
5553 ;READ UPDATE TO CACHE LOCATION 0000
5554 034434 042737 000100 177746 BIC #WVPD,CCR ;WRITE WRONG PARITY TO HI/LO BYTE PARITY STORES
5555 034442 005037 177744 CLR CME ;DISABLE WVPD
5556 034446 042737 000004 177746 BIC #FMLO,CCR ;CLEAR CME AND PARITY DETECT LOGIC
5557 034454 052737 000200 177746 BIS #PEA,CCR ;ENABLE LOW CACHE
5558 034462 011003 MOV (R0),R3 ;ALLOW FOR ABORT
5559 ;READ HIT:
5560 ;LO & HI BYTE PARITY CHECK GENERATORS WILL
; DETECT WRONG PARITY
```

```

5561                                     :USING HODO AND SOURCE MODE FOR READING
5562                                     :LOCATION 0 WILL INHIBIT PARITY ERROR
5563                                     :FROM BEING CLOCKED TO INTERRUPT LOGIC
5564                                     :HOWEVER, THE PARITY ERROR SIGNAL
5565                                     :WILL CAUSE THE ABORT SIGNAL TO BE
5566                                     :ASSERTED. THE ABORT SIGNAL WILL BE
5567                                     :CAUSE CME<15> TO BE SET.
5568                                     :THIS INSTRUCTION SHOULD BE ABORTED
5569 034464 000240      NOP
5570 034466 005237 002070      INC      FAIL1
5571 034472 012737 001015 177746      MOV      #OFF,CCR
5572 034500 000404      BR       5$
5573 034502 012737 001015 177746 4$:  MOV      #OFF,CCR
5574 034510 022626      CMP      (SP)+,(SP)+
5575 034512 013701 177744      5$:  MOV      CME,R1
5576 034516 005037 177744      CLR      CME
5577 034522 013702 177744      MOV      CME,R2
5578 034526 000240      25$:  NOP
5579 034530 000240      NOP
5579 034532 105037 177750      CLRB    CMR
5580 034536 012737 000002 000000      MOV      #2,0
5581 034544 012737 000116 000114      MOV      #116,114
5582 034552 005037 000116      CLR      116
5583 034556 052737 000400 177746      BIS      #FC,CCR
5584                                     :BEFORE LEAVING TEST FLUSH CACHE TO
5585 034564 032737 010000 177746 6$:  BIT      #VCIP,CCR
5586 034572 001374      BNE     6$
5587 034574 032701 100000      BIT      #CMPE,R1
5588 034600 001003      BNE     7$
5589 034602 104413      ERROR
5590                                     :-----
5590 034604 034602      .WORD  -2
5591                                     :INTERRUPT/ABORT LOGIC
5592                                     :CME<15> WAS NOT SET DUE TO ABORT
5593                                     :SIGNAL
5593 034606 000000      .WORD  0
5594 034610 032702 100000      7$:  BIT      #CMPE,R2
5595 034614 001403      BEQ     8$
5596 034616 104413      ERROR
5597                                     :-----
5597 034620 034616      .WORD  -2
5598                                     :INTERRUPT/ABORT TESTS
5599                                     :CME<15> WAS NOT CLEARED BY WRITE TO
5600                                     : TO CME
5600 034622 000000      .WORD  0
5601 034624 022703 177777      8$:  CMP      #-1,R3
5602 034630 001403      BEQ     9$
5603 034632 104413      ERROR
5604                                     :-----
5604 034634 034632      .WORD  -2
5605                                     :INTERRUPT/ABORT TESTS
5606                                     :R3 WAS OVERWRITTEN WITH DATA INDICATING
5607                                     :THAT INSTRUCTION WAS NOT ABORTED
5607 034636 000000      .WORD  0
5608 034640 005737 002070      9$:  TST     FAIL1
5609 034644 001403      BEQ     10$
5610 034646 104413      ERROR
    
```

5611 034650 034646
5612
5613 034652 000000
5614 034654 000240
034656 005237 001472

10\$: .WORD -2
.WORD 0
NOP
INC \$TESTN

:-----
:INTERRUPT/ABORT TESTS
:TRAP DID NOT OCCUR DUE TO ABORT
:END OF TEST
:INCREMENT TEST COUNTER

5629

```
.SBTTL TEST # 176 - CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S
:*****
:*TEST 176      CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S
:*      VERIFY CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S BY PERFORMING A
:*      MARCH PATTERN TEST TO LOW CACHE AREA OF TAG/DATA PARITY STORE (LOC. 0000-3777)
:*      PROCEDURE: 1. WRITE ALL 0'S TO ALL LO CACHE TAG/DATA PARITY STORE
:*                  RAMS CORRESPONDING TO LOCATIONS 0000-3777
:*                  2. READ 0'S FROM ALL LO CACHE RAMS CORRESPONDING
:*                     TO LOCATION 0000
:*                  3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
:*                     0000.
:*                  4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
:*                     0000.
:*                  5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
:*                     AND UNTIL LOC. 3777 IS REACHED.
:*****
```

```
TST176:
034662      000004      SPCPCND      ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
034662      000004      ;ERROR/LOOP ON TEST
034664      034674      .WORD      40$      ;TEST START LOCATION
034666      070000      .WORD      1$-40$+67764 ;LOOP ON ERROR START LOCATION
034670      000000      .WORD      0      ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
034672      070126      .WORD      25$-40$+67764 ;LOOP ON ERROR END LOCATION
034674      012737      001015 177746 40$: MOV      #OFF,CCR ;DISABLE CACHE
034702      004437      002452 JSR      R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
034706      035310      .WORD      10$+2 ;ADDRESS OF START OF NEXT TEST
```

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

```
5630 034710 012702 000401 1$: MOV      #401,R2 ;SETUP R2 WITH PATTERN 401
5631 034714 005037 177752 CLR      CHR ;LOAD AMR<8:0> WITH ALL 0'S
5632 034720 012700 060000 MOV      #60000,R0 ;ADDRESS 60000 TO R0
5633 034724 005020 2$: CLR      (R0)+ ;CLEAR ALL LOW CACHE MAIN MEMORY
5634 034726 020027 070000 CMP      R0,#70000
5635 034732 001374 BNE      2$
5636 034734 012700 060000 MOV      #60000,R0 ;1ST ADDRESS LOCATION IN R0
5637 034740 012701 040000 MOV      #40000,R1 ;ADDRESS 40000 IN R1
5638 034744 112737 000003 177750 MOVB     #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & DATA /TAG PARITY
5639 ; STORE BITS TO BE WRITTEN TO
5640 ;CMR<11:9> ONLY DURING THE DESTINATION
5641 ;ACCESS OF AN INSTRUCTION.
5642 ;TDAR ALLOWS TAG PARITY STORE GENERATOR
5643 ;INPUTS TO BE LOADED FROM AMR<8:0>
5644 034752 012737 000015 177746 MOV      #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
5645 034760 005721 6$: TST     (R1)+ ;
5646 034762 005720 TST     (R0)+ ;WRITE 0'S INTO ALL DATA/TAG PARITY STORE
5647 ;ADDRESS LOCATIONS SPECIFIED BY R0
5648 034764 022700 070000 CMP      #70000,R0 ;DONE?
5649 034770 001373 BNE      6$ ;NO
5650 034772 012737 000001 177752 MOV      #1,CHR ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
5651 035000 012700 060000 MOV      #60000,R0 ;ADDR. 60000 TO R0
5652 035004 042737 000001 177750 7$: BIC     #TDAR,CMR ;DISABLE TDAR TO ALLOW UPDATE
5653 ;OF CACHE TAG STORE THRU CA<21:13>
5654 035012 005710 TST     (R0) ;READ MISS TO CACHE LOCATION SPECIFIED
5655 ;BY R0. CLOCK TAG/DATA PARITY STORE
5656 ;BITS INTO CMR<11:9>.SHOULD BE ALL 0'S.
```


5753

```
.SBTTL TEST # 177 - CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S
*****
*TEST 177      CHK CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S
*   VERIFY CACHE TAG/DATA PARITY STORE RAM MEMORY IC'S BY PERFORMING A
*   MARCH PATTERN TEST TO HIGH CACHE AREA OF TAG/DATA PARITY STORE(LOC. 4000-7777)
*   PROCEDURE: 1. WRITE ALL 0'S TO ALL HI CACHE TAG/DATA PARITY STORE
*               RAMS CORRESPONDING TO LOCATIONS 4000-7777
*               2. READ 0'S FROM ALL HI CACHE RAMS CORRESPONDING
*               TO LOCATION 4000
*               3. WRITE ALL 1'S TO ALL RAMS CORRESPONDING TO LOCATION
*               4000
*               4. READ 1'S FROM ALL RAMS CORRESPONDING TO LOCATION
*               4000
*               5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
*               AND UNTIL LOC. 7777 IS REACHED.
*****
```

```
035314
035314 000004          SPCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
035316 035326          .WORD 40$          ;TEST START LOCATION
035320 060000          .WORD 1$-40$+57764      ;LOOP ON ERROR START LOCATION
035322 000000          .WORD 0          ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
035324 060126          .WORD 25$-40$+57764  ;LOOP ON ERROR END LOCATION
035326 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
035334 004437 002424 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
035340 035742          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST
```

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

```
5754 035342 012702 000401 1$: MOV #401,R2 ;SETUP R2 WITH PATTERN 401
5755 035346 005037 177752 CLR CHR ;LOAD AMR<8:0> WITH ALL 0'S
5756 035352 012700 070000 MOV #70000,R0 ;ADDRESS 70000 TO R0
5757 035356 005020 2$: CLR (R0)+ ;CLEAR ALL HIGH CACHE MAIN MEMORY
5758 035360 020027 100000 CMP R0,#100000
5759 035364 001374 BNE 2$
5760 035366 012700 070000 MOV #70000,R0 ;1ST ADDRESS LOCATION IN R0
5761 035372 012701 050000 MOV #50000,R1 ;ADDRESS 50000 IN R1
5762 035376 112737 000003 177750 MOVB #HODO+TDAR,CMR ;HODO ALLOWS CACHE UPDATES & DATA /TAG PARITY
5763 ; STORE BITS TO BE WRITTEN TO
5764 ;CMR<11:9> ONLY DURING THE DESTINATION
5765 ;ACCESS OF AN INSTRUCTION.
5766 ;TDAR ALLOWS TAG PARITY STORE GENERATOR
5767 ;INPUTS TO BE LOADED FROM AMR<8:0>
5768 035404 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE ENABLE PARITY STORE
5769 035412 005721 6$: TST (R1)+
5770 035414 005720 TST (R0)+
5771 ;WRITE 0'S INTO ALL DATA/TAG PARITY STORE
5772 035416 022700 100000 CMP #100000,R0 ;ADDRESS LOCATIONS SPECIFIED BY R0
5773 035422 001373 BNE 6$ ;DONE?
5774 035424 012737 000001 177752 MOV #1,CHR ;NO
5775 035432 012700 070000 MOV #70000,R0 ;LOAD AMR<8:0> BY WRITING TO CHR<8:0>
5776 035436 042737 000001 177750 7$: BIC #TDAR,CMR ;ADDR. 70000 TO R0
5777 ;DISABLE TDAR TO ALLOW UPDATE
5778 035444 005710 TST (R0) ;OF CACHE TAG STORE THRU CA<21:13>
5779 ;READ MISS TO CACHE LOCATION SPECIFIED
5780 ;BY R0. CLOCK TAG/DATA PARITY STORE
;BITS INTO CMR<11:9>.SHOULD BE ALL 0'S.
```

5781										:ALSO CAUSES UPDATE TO CACHE.TAG/DATA
5782										:PARITY STORE LOCATION SHOULD REMAIN
5783										:WITH 0'S.
5784	035446	013705	177750					MOV	CMR,R5	:SAVE CMR CONTENTS
5785	035452	052737	000J01	177750				BIS	#TDAR,CMR	:ENABLE TDAR TO ALLOW TAG PARITY GENERATOR
5786										:INPUTS TO SEE ODD DATA FROM AMR<8:0>
5787	035460	010210						MOV	R2,(R0)	:WRITE HIT CAUSES UPDATE TO CACHE.
5788										:TAG/DATA PARITY STORES WILL BE WRITTEN
5789										:WITH 1'S DUE TO AMR<8:0> ODD DATA
5790										:AND PATTERN 401 FROM R2 BEING PUT
5791										:ONTO PAX DATA LINES RESULTING IN
5792										:ODD DATA FOR LO AND HI BYTE DATA PARITY
5793										:GENERATORS.
5794	035462	005710						TST	(R0)	:READ MISS.CLOCK TAG/DATA PARITY STORE BITS TO
5795										:CMR<11:9>.SHOULD BE ALL 1'S.
5796	035464	013703	177750					MOV	CMR,R3	:SAVE CMR CONTENTS
5797	035470	000240			25\$:			NOP		:INSTRUCTION 'JMP 1\$' PLACED HERE
	035472	000240						NOP		:FOR LOOP ON ERROR
5798	035474	042705	170777					BIC	#170777,R5	:INTERESTED IN BITS 11:9
5799	035500	005705						TST	R5	:BITS 11:9 SHOULD BE ALL 0'S
5800	035502	001437						BEQ	8\$:PASS
5801	035504	012737	001015	177746				MOV	#OFF,CCR	:DISABLE CACHE
5802	035512	105037	177750					CLRB	CMR	:CLEAR MAINT. MODE
5803	035516	005037	050510					CLR	EXDAT1	:SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
5804	035522	010537	050526					MOV	R5,CMR119	:SPECIFY CACHE TAG/DATA PARITY STORE DATA RECEIVED
5805										:THRU CMR<11:9>
5806	035526	012737	000011	002062				MOV	#9,LOOP	:PREPARE CMR119 FOR TYPEOUT
5807	035534	006237	050526		11\$:			ASR	CMR119	
5808	035540	042737	100000	050526				BIC	#100000,CMR119	
5809	035546	005337	002062					DEC	LOOP	
5810	035552	001370						BNE	11\$	
5811	035554	010037	050506					MOV	R0,CA121	:SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
5812	035560	006237	050506					ASR	CA121	
5813	035564	104413						ERROR		:ERROR
	035566	035564						.WORD	.-2	:-----
5814										:TAG/DATA PARITY STORE MARCH PATTERN TEST
5815										:READING CACHE TAG/DATA PARITY STORE DATA
5816										:THRU CMR<11:9> DID NOT READ ALL 0'S.
5817										:THIS SUGGESTS THAT A RAM LOCATION
5818										:SPECIFIED BY CA121 WAS OVERWRITTEN
5819										:WITH A 1 WHEN WRITING A 1 TO ANOTHER
5820										:LOCATION.ANY BIT IN CMR119 DATA
5821										:THAT IS A 1 MAY POINT TO A BAD
5822										:CACHE TAG/DATA PARITY STORE RAM.
5823										
5824	035570	050510						EXDAT1		:PRINT EXPECTED CACHE TAG/DATA PARITY STORE DATA
5825	035572	050526						CMR119		:PRINT CACHE TAG/DATA PARITY STORE RECEIVED
5826										:THRU CMR<11:9>
5827	035574	050506						CA121		:SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
5828	035576	000000						.WORD	0	
5829	035600	000452						BR	3\$:END THE TEST
5830	035602	042703	170777		8\$:			BIC	#170777,R3	:INTERESTED IN BITS 11:9 ONLY
5831	035606	022703	007000					CMR	#7000,R3	:BITS 11:9 SHOULD BE ALL 1'S
5832	035612	001440						BEQ	9\$:PASS
5833	035614	012737	001015	177746				MOV	#OFF,CCR	:DISABLE CACHE
5834	035622	105037	177750					CLRB	CMR	:CLEAR MAINT. MODE

```

5835 035626 012737 000007 050510      MOV      #7,EXDAT1      ;SPECIFY EXPECTED CACHE TAG/DATA PARITY STORE DATA
5836 035634 010337 050526      MOV      R3,CMR119     ;SPECIFY CACHE TAG/DATA PARITY STORE DATA READ
5837                                     ;THRU CMR<11:9>
5838 035640 012737 000011 002062      MOV      #9,LOOP      ;PREPARE CMR119 FOR TYPEOUT
5839 035646 006237 050526      ASR      CMR119
5840 035652 042737 100000 050526 12$:    BIC      #100000,CMR119
5841 035660 005337 002062      DEC      LOOP
5842 035664 001370      BNE      12$
5843 035666 010037 050506      MOV      R0,CA121     ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
5844 035672 006237 050506      ASR      CA121
5845 035676 104413      ERROR    ;ERROR
                                     ;-----
                                     .WORD   -2
5846                                     ;TAG/DATA PARITY STORE MARCH PATTERN TEST
5847                                     ;READING CACHE TAG/DATA PARITY STORE DATA
5848                                     ;THRU CMR<11:9> DID NOT READ ALL 1'S.
5849                                     ;ANY BIT IN CMR119 DATA
5850                                     ;THAT IS A 0 MAY POINT TO A BAD
5851                                     ;CACHE TAG/DATA PARITY STORE RAM.
5852
5853 035702 050510      EXDAT1   ;PRINT EXPECTED CACHE TAG/DATA PARITY STORE DATA
5854 035704 050526      CMR119  ;PRINT CACHE TAG/DATA PARITY STORE DATA READ
5855                                     ;THRU CMR<11:9>
5856 035706 050506      CA121   ;SPECIFY FAILED TAG/DATA PARITY STORE ADDRESS LOCATION
5857 035710 000000      .WORD   0
5858 035712 000405      BR      3$           ;END TEST
5859 035714 062700 000002 9$:    ADD      #2,R0       ;NEXT LOCATION
5860 035720 022700 100000      CMP      #100000,R0   ;HAS ALL HI CACHE BEEN DONE?
5861 035724 001244      BNE      7$           ;NO,CONTINUE
5862 035726 012737 001015 177746 3$:    MOV      #OFF,CCR     ;DISABLE CACHE
5863 035734 105037 177750      CLR     CMR          ;DISABLE MAINT. MODE
5864 035740 000240 10$:    NOP
035742 005237 001472      INC     $TESTN      ;INCREMENT TEST COUNTER
  
```

5879

```
.SBTTL TEST # 200 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
:*****
:TEST 200 VERIFY CACHE VALID STORE RAM MEMORY IC'S
: VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
: MARCH PATTERN TEST TO LOW CACHE AREA OF VALID STORE(LOC. 0000-3777)
: PROCEDURE: 1. WRITE 0'S TO LO CACHE VALID STORE
: RAM CORRESPONDING TO LOCATIONS 0000-3777
: 2. READ 0 FROM LO CACHE RAM CORRESPONDING
: TO LOCATION 0000
: 3. WRITE 1 TO RAM CORRESPONDING TO LOCATION
: 0000.
: 4. READ 1 FROM RAM CORRESPONDING TO LOCATION
: 0000.
: 5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
: AND UNTIL LOC. 3777 IS REACHED.
:*****
```

```
035746
035746 000004

035750 035760
035752 070026
035754 000000
035756 070136
035760 012737 001015 177746 40$:
035766 004437 002452
035772 036276
```

```
TST200:
          SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          .WORD 40$             ;TEST START LOCATION
          .WORD 1$-40$+67764   ;LOOP ON ERROR START LOCATION
          .WORD 0               ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
          .WORD 25$-40$+67764  ;LOOP ON ERROR END LOCATION
001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
002452          JSR R4,RELCTH   ;LOCATE TEST CODE TO HIGH CACHE SPACE
          .WORD 10$+2          ;ADDRESS OF START OF NEXT TEST
```

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

```
5880 035774 032737 020000 177746          BIT #VSIU,CCR ;IS SET A USED
5881 036002 001407                BEQ 1$         ;YES
5882 036004 052737 000400 177746          BIS #FC,CCR   ;FLUSH
5883 036012 032737 010000 177746 200$: BIT #VCIP,CCR
5884 036020 001374                BNE 200$
5885 036022 012700 060000          1$: MOV #60000,R0 ;1ST ADDRESS LOCATION IN R0
5886 036026 012701 040000          MOV #40000,R1 ;ADDRESS 40000 IN R1
5887 036032 005002                CLR R2
5888 036034 112737 000002 177750          MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & VALID
5889                ; STORE BITS TO BE WRITTEN TO
5890                ;CMR<12> ONLY DURING THE DESTINATION
5891                ;ACCESS OF AN INSTRUCTION.
5892                ;STORE
5893 036042 012737 000015 177746          MOV #15,CCR   ;NO UCB SO AS TO UPDATE VALID STORE
5894 036050 005721                TST (R1)+
5895 036052 005720                TST (R0)+
5896                ;
5897 036054 022700 070000          CMP #70000,R0 ;UPDATE ALL LO CACHE VALID STORE
5898 036060 001373                BNE 6$         ;ADDRESS LOCATIONS SPECIFIED BY R0
5899 036062 012700 060000          MOV #60000,R0 ;DONE?
5900 036066 052737 001000 177746          BIS #UCB,CCR  ;NO
5901 036074 010220          13$: MOV R2,(R0)+ ;ADDR. 60000 TO R0
5902                ;ENABLE UCB
5903 036076 022700 070000          CMP #70000,R0 ;WRITE HIT WITH UCB WILL INVALIDATE
5904 036102 001374                BNE 13$        ;OR WRITE 0 TO ALL LO CACHE VALID STORE
5905 036104 042737 001000 177746          BIC #UCB,CCR  ;DONE?
5906 036112 012700 060000          MOV #60000,R0 ;NO
                ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
                ;ADDRESS 60000 TO R0
```

5907	036116	005710		7\$:	TST	(R0)			:READ MISS TO CACHE LOCATION SPECIFIED
5908									:BY R0. CLOCK VALID STORE
5909									:BIT INTO CMR<12>. SHOULD BE 0.
5910									:ALSO CAUSES UPDATE TO CACHE.
5911									:VALID STORE LOCATION WILL BE WRITTEN WITH A 1.
5912	036120	013705	177750		MOV	CMR,R5			:SAVE CMR CONTENTS
5913	036124	005710			TST	(R0)			:READ HIT. CLOCK VALID STORE BIT TO CMR<12> SHOULD BE 1.
5914	036126	013703	177750		MOV	CMR,R3			:SAVE CMR CONTENTS
5915	036132	000240		25\$:	NOP				:INSTRUCTION 'JMP 1\$' PLACED HERE
	036134	000240			NOP				:FOR LOOP ON ERROR
5916	036136	042705	167777		BIC	#167777,R5			:INTERESTED IN BIT 12
5917	036142	005705			TST	R5			:BIT 12 SHOULD BE 0
5918	036144	001416			BEQ	8\$:PASS
5919	036146	012737	001015	177746	MOV	#OFF,CCR			:DISABLE CACHE
5920	036154	105037	177750		CLRB	CMR			:CLEAR MAINT. MODE
5921	036160	010037	050506		MOV	R0,CA121			:SPECIFY FAILED VALID STORE ADDRESS LOCATION
5922	036164	006237	050506		ASR	CA121			
5923	036170	104413			ERROR				:ERROR
									:-----
	036172	036170			.WORD	.-2			
5924									:VALID STORE MARCH PATTERN TEST- SET A
5925									:READING CACHE VALID STORE DATA
5926									:THRU CMR<12> DID NOT READ 0.
5927									:THIS SUGGESTS THAT A RAM LOCATION
5928									:SPECIFIED BY CA121 WAS OVERWRITTEN
5929									:WITH A 1 WHEN WRITING A 1 TO ANOTHER
5930									:LOCATION.
5931									:THIS INDICATES THAT VALID STORE RAM
5932									:SET A IS BAD.
5933									
5934	036174	050506			CA121				:SPECIFY FAILED VALID STORE ADDRESS LOCATION
5935	036176	000000			.WORD	0			
5936	036200	000430			BR	3\$:END THE TEST
5937	036202	042703	167777	8\$:	BIC	#167777,R3			:INTERESTED IN BIT 12 ONLY
5938	036206	022703	010000		CMR	#10000,R3			:BIT 12 SHOULD BE 1
5939	036212	001416			BEQ	9\$:PASS
5940	036214	012737	001015	177746	MOV	#OFF,CCR			:DISABLE CACHE
5941	036222	105037	177750		CLRB	CMR			:CLEAR MAINT. MODE
5942	036226	010037	050506		MOV	R0,CA121			:SPECIFY FAILED VALID STORE ADDRESS LOCATION
5943	036232	006237	050506		ASR	CA121			
5944	036236	104413			ERROR				:ERROR
									:-----
	036240	036236			.WORD	.-2			
5945									:VALID STORE MARCH PATTERN TEST- SET A
5946									:READING CACHE VALID STORE DATA
5947									:THRU CMR<12> DID NOT READ 1.
5948									:THIS SUGGESTS THAT VALID STORE RAM
5949									:IC SET A IS BAD.
5950									:THRU CMR<12>
5951	036242	050506			CA121				:SPECIFY FAILED VALID STORE ADDRESS LOCATION
5952	036244	000000			.WORD	0			
5953	036246	000405			BR	3\$:END TEST
5954	036250	062700	000002	9\$:	ADD	#2,R0			:NEXT LOCATION
5955	036254	022700	070000		CMR	#70000,R0			:HAS ALL LO CACHE BEEN DONE?
5956	036260	001316			BNE	7\$:NO,CONTINUE
5957	036262	012737	001015	177746	MOV	#OFF,CCR			:DISABLE CACHE
5958	036270	105037	177750	3\$:	CLRB	CMR			:DISABLE MAINT. MODE

5959 036274 000240
036276 005237 001472

10\$: NOP
 INC \$TESTN

:END OF TEST
:INCREMENT TEST COUNTER

5974

```

.SBTTL TEST # 201 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
*TEST 201 VERIFY CACHE VALID STORE RAM MEMORY IC'S
*   VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
*   MATCH 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.
*****
  
```

```

036302
036302 000004

036304 036314
036306 060026
036310 000000
036312 060136
036314 012737 001015 177746
036322 004437 002424
036326 036632
  
```

```

TST201:
      SCPCND                ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                          ;ERROR/LOOP ON TEST
                          ;TEST START LOCATION
      .WORD 40$            ;LOOP ON ERROR START LOCATION
      .WORD 1$-40$+57764  ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
      .WORD 0              ;LOOP ON ERROR END LOCATION
      .WORD 25$-40$+57764 ;DISABLE CACHE
      MOV #OFF,CCR        ;LOCATE TEST CODE TO LOW CACHE SPACE
      JSR R4,RELCTL       ;ADDRESS OF START OF NEXT TEST
      .WORD 10$+2
  
```

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

```

5975 036330 032737 020000 177746      BIT #VSIU,CCR ;IS SET A USED
5976 036336 001407                BEQ 1$        ;YES
5977 036340 052737 000400 177746      BIS #FC,CCR  ;FLUSH
5978 036346 032737 010000 177746 200$: BIT #VCIP,CCR
5979 036354 001374                BNE 200$
5980 036356 012700 070000          1$: MOV #70000,R0 ;1ST ADDRESS LOCATION IN R0
5981 036362 012701 050000          MOV #50000,R1 ;ADDRESS 50000 IN R1
5982 036366 005002                CLR R2
5983 036370 112737 000002 177750      MOVB #HODO,LMR ;HODO ALLOWS CACHE UPDATES & VALID
5984                                ; STORE BITS TO BE WRITTEN TO
5985                                ;CMR<12> ONLY DURING THE DESTINATION
5986                                ;ACCESS OF AN INSTRUCTION. STORE
5987 036376 012737 000015 177746      MOV #15,CCR  ;NO UCB SO AS TO UPDATE VALID STORE
5988 036404 005721                TST (R1)+
5989 036406 005720                TST (R0)+
5990                                ;UPDATE ALL HI CACHE VALID STORE
5991 036410 022700 100000          CMP #100000,R0 ;ADDRESS LOCATIONS SPECIFIED BY R0
5992 036414 001373                BNE 6$
5993 036416 012700 070000          MOV #70000,R0 ;DONE?
5994 036422 052737 001000 177746      BIS #UCB,CCR ;NO
5995 036430 010220                MOV R2,(R0)+ ;ADDR. 70000 TO R0
5996                                ;ENABLE UCB
5997 036432 022700 100000          CMP #100000,R0 ;WRITE HIT WITH UCB WILL INVALIDATE
5998 036436 001374                BNE 13$      ;OR WRITE 0 TO ALL HI CACHE VALID STORE
5999 036440 042737 001000 177746      BIC #UCB,CCR ;DONE?
6000 036446 012700 070000          MOV #70000,R0 ;NO
6001 036452 005710                TST (R0)    ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
                          ;ADDRESS 70000 TO R0
                          ;READ MISS TO CACHE LOCATION SPECIFIED
  
```

6002									:BY R0. CLOCK VALID STORE
6003									:BIT INTO CMR<12>. SHOULD BE 0.
6004									:ALSO CAUSES UPDATE TO CACHE.
6005									:VALID STORE LOCATION WILL BE WRITTEN WITH A 1.
6006	036454	013705	177750						:SAVE CMR CONTENTS
6007	036460	005710							:READ HIT. CLOCK VALID STORE BIT TO CMR<12>
6008									:SHOULD BE 1.
6009	036462	013703	177750						:SAVE CMR CONTENTS
6010	036466	000240		25\$:					:INSTRUCTION 'JMP 1\$' PLACED HERE
	036470	000240							:FOR LOOP ON ERROR
6011	036472	042705	167777						:INTERESTED IN BIT 12
6012	036476	005705							:BIT 12 SHOULD BE 0
6013	036500	001416							:PASS
6014	036502	012737	001015	177746					:DISABLE CACHE
6015	036510	105037	177750						:CLEAR MAINT. MODE
6016	036514	010037	050506						:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6017	036520	006237	050506						:ERROR
6018	036524	104413							:-----
	036526	036524							:VALID STORE MARCH PATTERN TEST- SET A
6019									:READING CACHE VALID STORE DATA
6020									:THRU CMR<12> DID NOT READ 0.
6021									:THIS SUGGESTS THAT A RAM LOCATION
6022									:SPECIFIED BY CA121 WAS OVERWRITTEN
6023									:WITH A 1 WHEN WRITING A 1 TO ANOTHER
6024									:LOCATION.
6025									:THIS INDICATES THAT VALID STORE RAM SET A IS BAD.
6026									:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6027	036530	050506							:END THE TEST
6028	036532	000000							:INTERESTED IN BIT 12 ONLY
6029	036534	000430							:BIT 12 SHOULD BE 1
6030	036536	042703	167777	8\$:					:PASS
6031	036542	022703	010000						:DISABLE CACHE
6032	036546	001416							:CLEAR MAINT. MODE
6033	036550	012737	001015	177746					:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6034	036556	105037	177750						:ERROR
6035	036562	010037	050506						:-----
6036	036566	006237	050506						:VALID STORE MARCH PATTERN TEST- SET A
6037	036572	104413							:READING CACHE VALID STORE DATA
	036574	036572							:THRU CMR<12> DID NOT READ 1.
6038									:THIS SUGGESTS THAT VALID STORE RAM
6039									:IC SET A IS BAD.
6040									:THRU CMR<12>
6041									:SPECIFY FAILED VALID STORE ADDRESS LOCATION
6042									:END TEST
6043									:NEXT LOCATION
6044	036576	050506							:HAS ALL HI CACHE BEEN DONE?
6045	036600	000000							:NO. CONTINUE
6046	036602	000405							:DISABLE CACHE
6047	036604	062700	000002	9\$:					:DISABLE MAINT. MODE
6048	036610	022700	100000						:END OF TEST
6049	036614	001316							:INCREMENT TEST COUNTER
6050	036616	012737	001015	177746	3\$:				
6051	036624	105037	177750						
6052	036630	000240			10\$:				
	036632	005237	001472						

6067

```

.SBTTL TEST # 202 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
TEST 202 VERIFY CACHE VALID STORE RAM MEMORY IC'S
VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
MARCH PATTERN TEST TO LOW CACHE AREA OF VALID STORE(LOC. 0000-3777)
PROCEDURE: 1. WRITE 0'S TO LO CACHE VALID STORE
RAM CORRESPONDING TO LOCATIONS 0000-3777
2. READ 0 FROM LO CACHE RAM CORRESPONDING
TO LOCATION 0000
3. WRITE 1 TO RAM CORRESPONDING TO LOCATION
0000.
4. READ 1 FROM RAM CORRESPONDING TO LOCATION
0000.
5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
AND UNTIL LOC. 3777 IS REACHED.
*****
  
```

```

036636
036636 000004

036640 036650
036642 070026
036644 000000
036646 070135
036650 012737 001015 177746
036656 004437 002452
036662 037166
  
```

```

TST202: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
.WORD 40$ ;LOOP ON ERROR START LOCATION
.WORD 1$-40$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
.WORD 0 ;LOOP ON ERROR END LOCATION
.WORD 25$-40$+67764 ;DISABLE CACHE
MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE
JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST
.WORD 10$+2
  
```

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

```

6068 036664 032737 020000 177746 BIT #VSIU,CCR ;IS SET B USED
6069 036672 001007 BNE 1$ ;YES
6070 036674 052737 000400 177746 BIS #FC,CCR ;FLUSH
6071 036702 032737 010000 177746 200$: BIT #VCIP,CCR
6072 036710 001374 BNE 200$
6073 036712 012700 060000 1$: MOV #60000,R0 ;1ST ADDRESS LOCATION IN R0
6074 036716 012701 040000 MOV #40000,R1 ;ADDRESS 40000 IN R1
6075 036722 005002 CLR R2
6076 036724 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & VALID
6077 ; STORE BITS TO BE WRITTEN TO
6078 ;CMR<12> ONLY DURING THE DESTINATION
6079 ;ACCESS OF AN INSTRUCTION. STORE
6080 036732 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO UPDATE VALID STORE
6081 036740 005721 6$: TST (R1)+
6082 036742 005720 TST (R0)+
6083 ;UPDATE ALL LO CACHE VALID STORE
6084 036744 022700 070000 CMP #70000,R0 ;ADDRESS LOCATIONS SPECIFIED BY R0
6085 036750 001373 BNE 6$ ;DONE?
6086 036752 012700 060000 MOV #60000,R0 ;NO
6087 036756 052737 001000 177746 BIS #UCB,CCR ;ADDR. 60000 TO R0
6088 036764 010220 13$: MOV R2,(R0)+ ;ENABLE UCB
6089 ;WRITE HIT WITH UCB WILL INVALIDATE
6090 036766 022700 070000 CMP #70000,R0 ;OR WRITE 0 TO ALL LO CACHE VALID STORE
6091 036772 001374 BNE 13$ ;DONE?
6092 036774 042737 001000 177746 BIC #UCB,CCR ;NO
6093 037002 012700 060000 MOV #60000,R0 ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
6094 037006 005710 7$: TST (R0) ;ADDRESS 60000 TO R0
;READ MISS TO CACHE LOCATION SPECIFIED
  
```

```

6095 ;BY RO. CLOCK VALID STORE
6096 ;BIT INTO CMR<12>. SHOULD BE 0.
6097 ;ALSO CAUSES UPDATE TO CACHE.
6098 ;VALID STORE LOCATION WILL BE WRITTEN WITH A 1.
6099 037010 013705 177750 MOV CMR,R5
6100 037014 005710 TST (R0)
6101 ;SAVE CMR CONTENTS
6102 037010 013703 177750 MOV CMR,R3
6103 037022 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
        037024 000240 NOP ;FOR LOOP ON ERROR
6104 037026 042705 167777 BIC #167777,R5 ;INTERESTED IN BIT 12
6105 037032 005705 TST R5 ;BIT 12 SHOULD BE 0
6106 037034 001416 BEQ 8$ ;PASS
6107 037036 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
6108 037044 105037 177750 CLRB CMR ;CLEAR MAINT. MODE
6109 037050 010037 050506 MOV RO,CA121 ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6110 037054 006237 050506 ASR CA121
6111 037060 104413 ERROR ;ERROR
        ;-----
        037062 037060 .WORD .-2
6112 ;VALID STORE MARCH PATTERN TEST- SET B
6113 ;READING CACHE VALID STORE DATA
6114 ;THRU CMR<12> DID NOT READ 0.
6115 ;THIS SUGGESTS THAT A RAM LOCATION
6116 ;SPECIFIED BY CA121 WAS OVERWRITTEN
6117 ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
6118 ;LOCATION.
6119 ;THIS INDICATES THAT VALID STORE RAM
6120 ;SET B IS BAD.
6121 ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6122 037064 050506 CA121
6123 037066 000000 .WORD 0
6124 037070 000430 BR 3$ ;END THE TEST
6125 037072 042703 167777 8$: BIC #167777,R3 ;INTERESTED IN BIT 12 ONLY
6126 037076 022703 010000 CMP #10000,R3 ;BIT 12 SHOULD BE 1
6127 037102 001416 BEQ 9$ ;PASS
6128 037104 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
6129 037112 105037 177750 CLRB CMR ;CLEAR MAINT. MODE
6130 037116 010037 050506 MOV RO,CA121 ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6131 037122 006237 050506 ASR CA121
6132 037126 104413 ERROR ;ERROR
        ;-----
        037130 037126 .WORD .-2
6133 ;VALID STORE MARCH PATTERN TEST- SET B
6134 ;READING CACHE VALID STORE DATA
6135 ;THRU CMR<12> DID NOT READ 1.
6136 ;THIS SUGGESTS THAT VALID STORE RAM
6137 ;IC SET B IS BAD.
6138 ;THRU CMR<12>
6139 ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6140 037132 050506 CA121
6141 037134 000000 .WORD 0
6142 037136 000405 BR 3$ ;END TEST
6143 037140 062700 000002 9$: ADD #2,R0 ;NEXT LOCATION
6144 037144 022700 070000 CMP #70000,R0 ;HAS ALL LO CACHE BEEN DONE?
6145 037150 001316 BNE 7$ ;NO, CONTINUE
6146 037152 012737 001015 177746 3$: MOV #OFF,CCR ;DISABLE CACHE

```

6147	037160	105037	177750		CLRB	CMR	.DISABLE MAINT. MODE
6148	037164	000240		10%:	NOP		:END OF TEST
	037166	005237	001472		INC	\$TESTN	:INCREMENT TEST COUNTER

6163

```
.SBTTL TEST # 203 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
*****
*TEST 203 VERIFY CACHE VALID STORE RAM MEMORY IC'S
* VERIFY CACHE VALID STORE RAM MEMORY IC'S BY PERFORMING A
* MARCH PATTERN TEST TO HIGH CACHE AREA OF VALID STORE(LOC. 4000-7777)
* PROCEDURE: 1. WRITE 0'S TO HI CACHE VALID STORE
* RAM CORRESPONDING TO LOCATIONS 4000-7777
* 2. READ 0 FROM HI CACHE RAM CORRESPONDING
* TO LOCATION 4000
* 3. WRITE 1 TO RAM CORRESPONDING TO LOCATION
* 4000
* 4. READ 1 FROM RAM CORRESPONDING TO LOCATION
* 4000
* 5. REPEAT SREPS 2 THRU 4 WITH NEXT LOCATION
* AND UNTIL LOC. 7777 IS REACHED.
*****
```

```
TST203:
037172 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
037172 000004 ;ERROR/LOOP ON TEST
037174 037204 .WORD 40$ ;TEST START LOCATION
037176 060026 .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
037200 000000 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
037202 060136 .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
037204 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
037212 004437 002424 JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
037216 037522 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
```

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

```
6164 037220 032737 020000 177746 BIT #VSIU,CCR ;IS SET B USED
6165 037226 001007 BNE 1$ ;YES
6166 037230 052737 000400 177746 BIS #FC,CCR ;FLUSH
6167 037236 032737 010000 177746 200$: BIT #VCJP,CCR
6168 037244 001374 BNE 200$
6169 037246 012700 070000 1$: MOV #70000,R0 ;1ST ADDRESS LOCATION IN R0
6170 037252 012701 050000 MOV #50000,R1 ;ADDRESS 50000 IN R1
6171 037256 005002 CLR R2
6172 037260 112737 000002 177750 MOVB #HODO,CMR ;HODO ALLOWS CACHE UPDATES & VALID
6173 ; STORE BITS TO BE WRITTEN TO
6174 ;CMR<12> ONLY DURING THE DESTINATION
6175 ;ACCESS OF AN INSTRUCTION.
6176 ;STORE
6177 037266 012737 000015 177746 6$: MOV #15,CCR ;NO UCB SO AS TO UPDATE VALID STORE
6178 037274 005721 TST (R1)+ ;
6179 037276 005720 TST (R0)+ ;
6180 ;UPDATE ALL HI CACHE VALID STORE
6181 037300 022700 100000 CMP #100000,R0 ;ADDRESS LOCATIONS SPECIFIED BY RC
6182 037304 001373 BNE 6$ ;DONE?
6183 037306 012700 070000 MOV #70000,R0 ;NO
6184 037312 052737 001000 177746 BIS #UCB,CCR ;ADDR. 70000 TO R0
6185 037320 010220 13$: MOV R2,(R0)+ ;ENABLE UCB
6186 ;WRITE HIT WITH UCB WILL INVALIDATE
6187 037322 022700 100000 CMP #100000,R0 ;OR WRITE 0 TO ALL HI CACHE VALID STORE
6188 037326 001374 BNE 13$ ;DONE?
6189 037330 042737 001000 177746 BIC #UCB,CCR ;NO
6190 037336 012700 070000 MOV #70000,R0 ;DISABLE UCB SO AS TO ALLOW VALID STORE UPDATES
;ADDRESS 70000 TO R0
```

```

6191 037342 005710          7$:  TST      (R0)      ;READ MISS TO CACHE LOCATION SPECIFIED
6192                                     ;BY R0.  CLOCK VALID STORE
6193                                     ;BIT INTO CMR<12>. SHOULD BE 0.
6194                                     ;ALSO CAUSES UPDATE TO CACHE.
6195                                     ;VALID STORE LOCATION WILL BE WRITTEN
6196                                     ;WITH A 1.
6197 037344 013705 177750      MOV      CMR,R5      ;SAVE CMR CONTENTS
6198 037350 005710          TST      (R0)      ;READ HIT. CLOCK VALID STORE BIT TO CMR<12>
6199                                     ;SHOULD BE 1.
6200 037352 013703 177750      MOV      CMR,R3      ;SAVE CMR CONTENTS
6201 037356 000240          25$:  NOP      ;INSTRUCTION 'JMP 1$' PLACED HERE
        037360 000240      NOP      ;FOR LOOP ON ERROR
6202 037362 042705 167777      BIC      #167777,R5 ;INTERESTED IN BIT 12
6203 037366 005705          IST      R5         ;BIT 12 SHOULD BE 0
6204 037370 001416          BEQ      8$         ;PASS
6205 037372 001015 177746      MOV      #OFF,CCR   ;DISABLE CACHE
6206 037400 105037 177750      CLR      CMR       ;CLEAR MAINT. MODE
6207 037404 010037 050506      MOV      R0,CA121  ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6208 037410 006237 050506      ASR      CA121
6209 037414 104413          ERROR      ;ERROR
        037416 037414          .WORD     .-2      ;-----

6210                                     ;VALID STORE MARCH PATTERN TEST- SET B
6211                                     ;READING CACHE VALID STORE DATA
6212                                     ;THRU CMR<12> DID NOT READ 0.
6213                                     ;THIS SUGGESTS THAT A RAM LOCATION
6214                                     ;SPECIFIED BY CA121 WAS OVERWRITTEN
6215                                     ;WITH A 1 WHEN WRITING A 1 TO ANOTHER
6216                                     ;LOCATION.
6217                                     ;THIS INDICATES THAT VALID STORE RAM
6218                                     ;SET B IS BAD.
6219
6220 037420 050506          CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6221 037422 000000          .WORD     0
6222 037424 000430          BR       3$
6223 037426 042703 167777      8$:  BIC      #167777,R3 ;INTERESTED IN BIT 12 ONLY
6224 037432 022703 010000      CMP      #10000,R3 ;BIT 12 SHOULD BE 1
6225 037436 001416          BEQ      9$         ;PASS
6226 037440 012737 001015 177746      MOV      #OFF,CCR   ;DISABLE CACHE
6227 037446 105037 177750      CLR      CMR       ;CLEAR MAINT. MODE
6228 037452 010037 050506      MOV      R0,CA121  ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6229 037456 006237 050506      ASR      CA121
6230 037462 104413          ERROR      ;ERROR
        037464 037462          .WORD     .-2      ;-----

6231                                     ;VALID STORE MARCH PATTERN TEST- SET B
6232                                     ;READING CACHE VALID STORE DATA
6233                                     ;THRU CMR<12> DID NOT READ 1.
6234                                     ;THIS SUGGESTS THAT VALID STORE RAM
6235                                     ;IC SET B IS BAD.
6236                                     ;THRU CMR<12>
6237 037466 050506          CA121      ;SPECIFY FAILED VALID STORE ADDRESS LOCATION
6238 037470 000000          .WORD     0
6239 037472 000405          BR       3$
6240 037474 062700 000002      9$:  ADD      #2,R0    ;END TEST
6241 037500 022700 100000      CMP      #100000,R0 ;NEXT LOCATION
6242 037504 001316          BNE      7$         ;HAS ALL HI CACHE BEEN DONE?
        ;NO,CONTINUE
    
```

6243	037506	012737	001015	177746	3\$:	MOV	#OFF,CCR	:DISABLE CACHE
6244	037514	105037	177750			CLRB	CMR	:DISABLE MAINT. MODE
6245	037520	000240			10\$:	NOP		:END OF TEST
	037522	005237	001472			INC	\$TESTN	:INCREMENT TEST COUNTER

6266

```
.SBTTL TEST # 204 - VERIFY CACHE VALID STORE RAM MEMORY IC'S
:*****
:*TEST 204 VERIFY CACHE VALID STORE RAM MEMORY IC'S
:* VERIFY THAT THE CACHE HIT NAND GATE CAN INDICATE A READ
:* HIT CONDITION.
:* PROCEDURE: CREATE A READ HIT CONDITION TO LO CACHE
:* WITH LO CACHE ENABLED,AND VERIFY THAT
:* OUTPUT OF THE CACHE HIT NAND GATE READS 0
:* THRU CMR<8>.
:* CONDITIONS: INPUTS CACHE HIT NAND GATE:
:* COMPARE 1 =1
:* COMPARE 2 =1
:* COMPARE 3 =1
:* VALID =1
:* TAG PAR. ERR =1
:* HI BYTE PE =1
:* LO BYTE PE =1
:* MISS HI =1
:* MISS LO =1
:* BYPASS/WRITE =1
:* FAULT =1
:*****
```

037526
 037526 000004
 037530 037540
 037532 070004
 037534 070030
 037536 070040
 037540 012737 001015 177746
 037546 004437 002452
 037552 037644

```
TST204:
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 EPROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
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

6267 037554 005037 060000
 6268 037560 112737 000002 177750
 6269
 6270
 6271
 6272 037566 012737 000011 177746
 6273
 6274 037574 005737 040000
 6275 037600 005737 060000
 6276 037604 005737 060000
 6277
 6278
 6279 037610 013701 177750
 6280 037614 000240
 037616 000240
 6281 037620 012737 001015 177746
 6282 037626 032701 000400
 6283 037632 001403
 6284 037634 104413
 037636 037634

```
CLR 60000 ;ALL 0'S TO MAIN MEMORY LOC. 60000
MOV# #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
;CLOCKING OF OUTPUT OF CACHE HIT NAND
;GATE INTO CMR ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
MOV #11,CCR ;NO UCB SO AS TO WRITE CACHE STORES
;ENABLE LOW CACHE FOR A READ HIT
;
TST 40000 ;
TST 60000 ;READ UPDATE TO LOW CACHE LOACATION 0000
TST 60000 ;READ HIT; ALL INPUTS OF CACHE HIT NAND
;GATE ARE 1; CLOCK STATUS OF NAND GATE
;OUTPUT TO CMR<8>
MOV CMR,R1 ;SAVE CMR CONTENTS
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
MOV #OFF,CCR ;DISABLE CACHE
BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 0
BEQ 10$ ;PASS
ERROR ;ERROR
;-----
;WORD .-?
```

6285
6286
6287
6288 037640 000000
6289 037642 000240
 037644 005237 001472

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

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

;END OF TEST
;INCREMENT TEST COUNTER

6309

```

.SBTTL TEST # 205 - CHECK FORCE MISS LOGIC.
*****
*TEST 205 CHECK FORCE MISS LOGIC.
* CHECK FORCE MISS LOGIC.
* PROCEDURE: CREATE A READ HIT CONDITION TO HIGH CACHE
* WITH 'FORCE MISS HI' DISABLED AND 'FORCE MISS
* LO' ENABLED. VERIFY OUTPUT OF CACHE HIT NAND
* GATE READS A 0 THRU CMR<8>.
* CONDITIONS: INPUTS CACHE HIT NAND GATE:
* COMPARE 1 =1
* COMPARE 2 =1
* COMPARE 3 =-1
* VALID =-1
* TAG PAR. ERR =-1
* HI BYTE PE =1
* LO BYTE PE =-1
* MISS HI =1
* MISS LO =-1
* BYPASS/WRITE =-1
* FAULT =-1
*****
    
```

```

037650 TST205: SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
037650 000004 ;ERROR/LOOP ON TEST
037652 037662 .WORD 40$ ;TEST START LOCATION
037654 060004 .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
037656 060030 .WORD 20$-40$+57764 ;SCOPE SYNC LOCATION
037660 060040 .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
037662 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
037670 004437 002424 JSR R4,RELCAL ;LOCATE TEST CODE TO LOW CACHE SPACE
037674 037766 .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
    
```

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

```

6310 037676 005037 070000 CLR 70000 ;ALL 0'S TO MAIN MEMORY LOC. 70000
6311 037702 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6312 ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6313 ;GATE INTO CMR ONLY DURING THE DESTINATION
6314 ;ACCESS OF AN INSTRUCTION.
6315 037710 012737 000005 177746 MOV #5,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6316 ;ENABLE HI CACHE FOR A READ HIT
6317 ;DISABLE LO CACHE
6318 037716 005737 050000 TST 50000 ;
6319 037722 005737 070000 TST 70000 ;READ UPDATE TO HI CACHE LOCATION 4000
6320 037726 005737 070000 20$: TST 70000 ;READ HIT: ALL INPUTS OF CACHE HIT NAND
6321 ;GATE ARE 1; CLOCK STATUS OF NAND GATE
6322 ;OUTPUT TO CMR<8>
6323 037732 013701 177750 MOV CMR,R1 ;SAVE CMR CONTENTS
6324 037736 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
037740 000240 NOP ;FOR LOOP ON ERROR
6325 037742 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
6326 037750 032701 000400 BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 0
6327 037754 001403 BEQ 10$ ;PASS
6328 037756 104413 ERROR ;ERROR
;-----
037760 037756 .WORD -2
    
```

6329
6330
6331
6332 037762 000000
6333 037764 000240
037766 005237 001472

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

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

6353

```
.SBTTL TEST # 206 - CHK 'FORCE MISS LO' INHIBITS CACHE NAND GATE
:*****
:TEST 206      CHK 'FORCE MISS LO' INHIBITS CACHE NAND GATE
:  VERIFY THAT 'FORCE MISS LO' WILL INHIBIT CACHE HIT NAND GATE
:  FROM INDICATING A CACHE HIT.
:  PROCEDURE:   WITH 'FORCE MISS LO' ENABLED ATTEMPT A READ HIT TO LOW CACHE.
:               VERIFY THAT THE OUTPUT OF CACHE HIT NAND GATE
:               WILL READ AS A 1 THRU CMR<8>.
:  CONDITIONS: INPUTS CACHE HIT NAND GATE:
:               COMPARE 1   =1
:               COMPARE 2   =1
:               COMPARE 3   =1
:               VALID      =1
:               TAG PAR. ERR =1
:               HI BYTE PE  =1
:               LO BYTE PE  =1
:               MISS HI     =1
:               MISS LO     =0
:               BYPASS/WRITE =1
:               FAULT       =1
:*****
```

037772
 037772 000004
 037774 040004
 037776 070004
 040000 070030
 040002 070040
 040004 012737 001015 177746 40\$:
 040012 004437 002452
 040016 040110

```
TST206:
          SPCOND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
          ;ERROR/LOOP ON TEST
          .WORD 40$      ;TEST START LOCATION
          .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
          .WORD 20$-40$+67764 ;SCOPE SYNC. LOCATION
          .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
          MOV #OFF,CCR    ;DISABLE CACHE
          JSR R4,RELCTH   ;LOCATE TEST CODE TO HIGH CACHE SPACE
          .WORD 10$+2     ;ADDRESS OF START OF NEXT TEST
```

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

```
6354 040020 005037 060000          CLR 60000 ;ALL 0'S TO MAIN MEMORY LOC. 60000
6355 040024 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6356 ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6357 ;GATE INTO CMR ONLY DURING THE DESTINATION
6358 ;ACCESS OF AN INSTRUCTION.
6359 040032 012737 000015 177746 MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6360 ;DISABLE LOW CACHE
6361 040040 005737 040000          TST 40000 ;
6362 040044 005737 060000          TST 60000 ;READ UPDATE TO LOW CACHE LOACATION 0000
6363 040050 005737 060000          TST 60000 ;READ HIT;
6364 ;CLOCK STATUS OF NAND GATE
6365 ;OUTPUT TO CMR<8>
6366 040054 013701 177750          MOV CMR,R1 ;SAVE CMR CONTENTS
6367 040060 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
        040062 000240 NOP ;FOR LOOP ON ERROR
6368 040064 012737 001015 177746 MOV #OFF,CCR ;DISABLE CACHE
6369 040072 032701 000400          BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
6370 040076 001003          BNE 10$ ;PASS
6371 040100 104413          ERROR ;ERROR
        ;-----
        040102 040100          .WORD -2
6372 ;CACHE HIT TESTS
```

6373
6374
6375 040104 000000
6376 040106 000240
040110 005237 001472

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

:READING OUTPUT OF CACHE HIT NAND GATE
:THRU CMR<8> DID NOT RESULT IN A 1
:END OF TEST
:INCREMENT TEST COUNTER

6393

```
.SBTTL TEST # 207 - CHK 'TAG PARITY ERROR' INHIBITS CACHE NAND GATE
:*****
*TEST 207      CHK 'TAG PARITY ERROR' INHIBITS CACHE NAND GATE
*   VERIFY THAT 'TAG PARITY ERROR' WILL INHIBIT CACHE HIT NAND GATE
*   FROM INDICATING A CACHE HIT.
*   CONDITIONS:   INPUTS CACHE HIT NAND GATE:
*                   COMPARE 1      =1
*                   COMPARE 2      =1
*                   COMPARE 3      =-1
*                   VALID          =1
*                   TAG PAR. ERR   =0
*                   HI BYTE PE     =1
*                   LO BYTE PE     =1
*                   MISS HI        =1
*                   MISS LO        =1
*                   BYPASS/WRITE   =1
*                   FAULT          =1
:*****
```

040114
040114 000004

040116 040126
040120 070000
040122 070054
040124 070072
040126 012737 001015 177746
040134 004437 002452
040140 040274

```
TST207:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
        ;ERROR/LOOP ON TEST
        .WORD 40$ ;TEST START LOCATION
        .WORD 1$-40$+67764 ;LOOP ON ERROR START LOCATION
        .WORD 20$-40$+67764 ;SCOPE SYNC. LOCATION
        .WORD 25$-40$+67764 ;LOOP ON ERROR END LOCATION
40$:    MOV #OFF,CCR ;DISABLE CACHE
        JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
        .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
```

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

6394 040142 005037 000000
6395 040146 112737 000002 177750
6396
6397
6398 040154 012737 000015 177746
6399 040162 005737 000000
6400 040166 005737 040000
6401 040172 052737 002000 177746
6402 040200 005737 000000
6403
6404
6405
6406 040204 042737 002004 177746
6407 040212 005037 177744
6408 040216 005737 000000 20\$:
6409
6410
6411
6412
6413
6414 040222 013701 177750
6415 040226 012737 001015 177746
6416 040234 000240 25\$:
040236 000240
6417 040240 052737 000400 177746

```
1$:    CLR 0 ;0'S TO MAIN MEMORY LOCATION 0.
        MOV# #HODO,CMR ;HODO ALLOWS UPDATES AND CACHE HITS
        ; ONLY DURING THE DESTINATION ACCESS OF
        ; AN INSTRUCTION.
        MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
        TST 0 ;
        TST 40000 ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
        BIS #WWPT,CCR ;ALLOW WRITE WRONG PARITY TO TAGG PARITY STORE
        TST 0 ;READ UPDATE TO CACHE LOCATION 0000;
        ;ALL 0'S(EVEN DATA) WILL BE WRITTEN TO DATA/TAG STORES,
        ;0'S INTO DATA PARITY STORES,AND A 1
        ;INTO TAG PARITY STORE.
        BIC #WWPT+FMLO,CCR ;DISABLE WWPT;ENABLE LOW CACHE
        CLR CME ;CLEAR CME
        TST 0 ;READ HIT; ALL 0'S WILL BE PLACED ON INPUTS
        ;OF DATA PARITY ERROR CHECK PARITY GEN'S, BUT
        ;TAG PARITY CHECK GENERATOR WILL
        ;SEE ODD DATA DUE TO WRONG PARITY
        ;FROM PREVIOUS READ UPDATE.
        ;OUTPUT TO CMR<B>
        MOV CMR,R1 ;SAVE CMR CONTENTS
        MOV #OFF,CCR ;DISABLE CACHE
        NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
        NOP ;FOR LOOP ON ERROR
        BIS #FC,CCR ;BEFORE LEAVING TEST ELIMINATE EFFECT OF WWPT
```

```

6418 040246 032737 010000 177746 500$: BIT #VCIP,CCR ;WAIT TILL DONE
6419 040254 001374 BNE 500$
6420 040256 032701 000400 BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
6421 040262 001003 B'IE 10$ ;PASS
6422 040264 104413 ERROR ;ERROR
;-----
040266 040264 .WORD .-2 ;CACHE HIT TESTS
6423 ;READING OUTPUT OF CACHE HIT NAND GATE
6424 ;THRU CMR<8> DID NOT RESULT IN A 1
6425
6426 040270 000000 .WORD 0
6427 040272 000240 10$: NOP ;END OF TEST
040274 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
    
```


6444

```
.SBTTL TEST # 210 - CHK 'LO & HI BYTE PARITY ERROR' STOPS NAND GATE
*****
*TEST 210      CHK 'LO & HI BYTE PARITY ERROR' STOPS NAND GATE
*  VERIFY THAT 'LO & HI BYTE PARITY ERROR' WILL INHIBIT CACHE HIT NAND GATE
*  FROM INDICATING A CACHE HIT.
*  CONDITIONS:  INPUTS CACHE HIT NAND GATE:
*                COMPARE 1      =1
*                COMPARE 2      =1
*                COMPARE 3      =1
*                VALID          =1
*                TAG PAR. ERR    1
*                HI BYTE PE     0
*                LO BYTE PE     =0
*                MISS HI        =1
*                MISS LO        =1
*                BYPASS/WRITE   =1
*                FAULT          1
*****
```

```
040300          C40300 000004          SCPCND          ;SCOPE CONDITIONS:GO SFT UP FOR LOOP ON
;ERROR/LOOP ON TEST
040302 040312      .WORD 40$          ;TEST START LOCATION
040304 070000      .WORD 1$-40$+67764        ;LOOP ON ERROR START LOCATION
040306 070054      .WORD 20$-40$+67764    ;SCOPE SYNC. LOCATION
040310 070072      .WORD 25$-40$+67764    ;LOOP ON ERROR END LOCATION
040312 012737 001015 177746 40$:  MOV #OFF,CCR          ;DISABLE CACHE
040320 004437 002452      JSR R4,RELCTH        ;LOCATE TEST CODE TO HIGH CACHE SPACE
040324 040460      .WORD 10$+2          ;ADDRESS OF START OF NEXT TEST
```

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

```
6445 040326 005037 000000 1$: CLR 0          ;0'S TO MAIN MEMORY LOCATION 0.
6446 040332 112737 000002 177750 MOV#B #HODO,CMR    ;HODO ALLOWS UPDATES AND CACHE HITS
6447                                     ; ONLY DURING THE DESTINATION ACCESS OF
6448                                     ; AN INSTRUCTION.
6449 040340 012737 000015 177746 MOV #15,CCR        ;NO UCB SO AS TO WRITE CACHE STORES
6450 040346 005737 000000      TST 0          ;
6451 040352 005737 040000      TST 40000       ;UPDATE CACHE LOCATION 0000 WITH CORRECT PARITY STORAGE
6452 040356 052737 000100 177746 BIS #WWPD,CCR      ;ALLOW WRITE WRONG PARITY DATA TO LO
6453                                     ; & HI BYTE PARITY STORE.
6454 040364 005737 000000      TST 0          ;READ UPDATE TO CACHE LOCATION 0000;
6455                                     ; ALL 0'S WILL BE WRITTEN TO DATA/TAG STORES,
6456                                     ; 1'S INTG LO & HI BYTE DATA PARITY STORES, AND A 0
6457                                     ; INTO TAG PARITY STORE.
6458 040370 042737 000104 177746 BIC #WWPD+FMI0,CCR ;DISABLE WWPD;ENABLE LO CACHE
6459 040376 005037 177744      CLR CME         ;CLEAR CME
6460 040402 005737 000000 20$: TST 0          ;READ HIT; ALL 0'S(EVEN DATA) WILL BE
6461                                     ; PLACED ON INPUTS
6462                                     ; OF TAG PARITY ERROR CHECK PARITY GEN'S, BUT
6463                                     ; LO & HI BYTE PARITY CHECK GENERATORS WILL
6464                                     ; SEE ODD DATA DUE TO WRONG PARITY
6465                                     ; FROM PREVIOUS READ UPDATE.
6466                                     ; CLOCK STATUS OF NAND GATE TO
6467                                     ; OUTPUT TO CMR<8>
6468 040406 013701 177750      MOV CMP,R1        ;SAVE CMR CONTENTS
6469 040412 012737 001015 177746 MOV #OFF,CCR      ;DISABLE CACHE
```

6470	040420	000240		25\$:	NOP				:INSTRUCTION 'JMP 1\$' PLACED HERE
	040422	000240			NOP				:FOR LOOP ON ERROR
6471	040424	052737	000400	177746	BIS	#FC,CCR			:BEFORE LEAVING TEST ELIMINATE EFFECTS OF WARD
6472	040432	032737	010000	177746	500\$:	BIT	#VCIP,CCR		:WAIT TILL DONE
6473	040440	001374			BNE	500\$			
6474	040442	032701	000400		BIT	#HIT,R1			:WAS CACHE HIT SIGNAL A
6475	040446	001003			BNE	10\$:PASS
6476	040450	104413			ERROR				:ERROR
	040452	040450			.WORD	.-2			:-----
6477									:CACHE HIT TESTS
6478									:READING OUTPUT OF CACHE HIT NAND GATE
6479									:THRU CMR<8> DID NOT RESULT IN A 1
6480	040454	000000			.WORD	0			
6481	040456	000240		10\$:	NOP				:END OF TEST
	040460	005237	001472		INC	\$TESTN			:INCREMENT TEST COUNTER

6501

```
.SBTTL TEST # 211 - 'FORCE MISS HI' INHIBITS NAND FROM IND. CACHE HIT
*****
*TEST 211 'FORCE MISS HI' INHIBITS NAND FROM IND. CACHE HIT
*   VERIFY THAT 'FORCE MISS HI' WILL INHIBIT CACHE HIT NAND GATE
*   FROM INDICATING A CACHE HIT.
*   PROCEDURE: WITH 'FORCE MISS HI' ENABLED ATTEMPT A READ HIT TO HI CACHE.
*               VERIFY THAT THE OUTPUT OF CACHE HIT NAND GATE
*               WILL READ AS A 1 THRU CMR<8>.
*   CONDITIONS: INPUTS CACHE HIT NAND GATE:
*               COMPARE 1      =1
*               COMPARE 2      =1
*               COMPARE 3      =1
*               VALID          =1
*               TAG PAR. ERR   =1
*               HI BYTE PE     =1
*               LO BYTE PE     =1
*               MISS HI        =0
*               MISS LO        =1
*               BYPASS/WRITE   =1
*               FAULT          =1
*****
```

040464
040464 000004

040466 040476
040470 060004
040472 060030
040474 060040
040476 012737 001015 177746 40\$:
040504 004437 002424
040510 040602

```
TST211:
      SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
      ;ERROR/LOOP ON TEST
      .WORD 40$ ;TEST START LOCATION
      .WORD 1$-40$+57764 ;LOOP ON ERROR START LOCATION
      .WORD 20$-40$+57764 ;SCOPE SYNC LOCATION
      .WORD 25$-40$+57764 ;LOOP ON ERROR END LOCATION
      MOV #OFF,CCR ;DISABLE CACHE
      JSR R4,RELCTL ;LOCATE TEST CODE TO LOW CACHE SPACE
      .WORD 10$+2 ;ADDRESS OF START OF NEXT TEST
```

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

6502 040512 005037 070000
6503 040516 112737 000002 177750 1\$:
6504
6505
6506
6507 040524 012737 000015 177746
6508
6509 040532 005737 050000
6510 040536 005737 070000
6511 040542 005737 070000 20\$:
6512
6513 040546 013701 177750
6514 040552 000240 25\$:
040554 000240
6515 040556 012737 001015 177746
6516 040564 032701 000400
6517 040570 001003
6518 040572 104413

040574 040572
6519
6520

```
      CLR 70000 ;ALL 0'S TO MAIN MEMORY LOC. 70000
      MOV #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
      ;CLOCKING OF OUTPUT OF CACHE HIT NAND
      ;GATE INTO CMR ONLY DURING THE DESTINATION
      ;ACCESS OF AN INSTRUCTION.
      MOV #15,CCR ;NO UCB SO AS TO WRITE CACHE STORES
      ;DISABLE HI CACHE
      ;
      TST 50000 ;
      TST 70000 ;READ UPDATE HI CACHE LOCATION 4000
      TST 70000 ;READ HIT; CLOCK STATUS OF NAND GATE
      ;OUTPUT TO CMR<8>
      MOV CMR,R1 ;SAVE CMR CONTENTS
      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
      ;-----
      .WORD -2 ;CACHE HIT TESTS
      ;READING OUTPUT OF CACHE HIT NAND GATE
```

6521
6522 040576 000000
6523 040600 000240
040602 005237 001472

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

:THRU CMR<8> DID NOT RESULT IN A 1
:END OF TEST
:INCREMENT TEST COUNTER

6543

```

.SBTTL TEST # 212 - 'UNCONDITIONAL CACHE BYPASS' NO HIT CHECK
*****
*TEST 212 'UNCONDITIONAL CACHE BYPASS' NO HIT CHECK
*   VERIFY THAT AN 'UNCONDITIONAL CACHE BYPASS' WILL INHIBIT CACHE HIT NAND GATE
*   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
*                 MISS LO     =1
*                 BYPASS/WRITE =0
*                 FAULT      =1
*****
    
```

```

040606
040606 000004          SCPCND          ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
                                ;ERROR/LOOP ON TEST
                                ;TEST START LOCATION
040610 040620          .WORD 40$          ;LOOP ON ERROR START LOCATION
040612 070004          .WORD 1$-40$+67764 ;SCOPE SYNC. LOCATION
040614 070036          .WORD 20$-40$+67764 ;LOOP ON ERROR END LOCATION
040616 070054          .WORD 25$-40$+67764
040620 012737 001015 177746 40$: MOV #OFF,CCR ;DISABLE CACHE
040626 004437 002452 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
040632 040732          .WORD 10$+2      ;ADDRESS OF START OF NEXT TEST
    
```

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

```

6544 040634 005037 060000          CLR 60000 ;ALL 0'S TO MAIN MEMORY LOCATION 60000
6545 040640 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6546                                     ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6547                                     ;GATE INTO CMR ONLY DURING THE DESTINATION
6548                                     ;ACCESS OF AN INSTRUCTION.
6549 040646 012737 000011 177746          MOV #11,CCR ;NO UCB SO AS TO UPDATE CACHE STORES
6550                                     ;ENABLE LOW CACHE
6551 040654 005737 040000          TST 40000 ;
6552 040660 005737 060000          TST 60000 ;READ UPDATE TO LOW CACHE LOCATION 0000
6553 040664 052737 001000 177746          BIS #UCB,CCR ;ENABLE UCB
6554 040672 005737 060000          20$: TST 60000 ;READ HIT; CLOCK STATUS OF NAND GATE
6555                                     ;OUTPUT TO CMR<8>
6556 040676 013701 177750          MOV CMR,R1 ;SAVE CMR CONTENTS
6557 040702 012737 001015 177746          MOV #OFF,CCR ;DISABLE CACHE
6558 040710 000240 25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
                                NOP ;FOR LOOP ON ERROR
6559 040714 032701 000400          BIT #HIT,R1 ;WAS CACHE HIT SIGNAL A 1
6560 040720 001003          BNE 10$ ;PASS
6561 040722 104413          ERROR ;ERROR
                                ;-----
040724 040722          .WORD -2 ;CACHE HIT TESTS
6562
    
```

6563
6564
6565 040726 000600
6566 040730 000240
040732 005237 001472

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

;READING OUTPUT OF CACHE HIT NAND GATE
;THRU CMR<8> DID NOT RESULT IN A 1
;END OF TEST
;INCREMENT TEST COUNTER

6591

```
.SBTTL TEST # 213 - 'VALID' INPUT TO NAND INHIBITS IND. CACHE HIT
:*****
:*TEST 213 'VALID' INPUT TO NAND INHIBITS IND. CACHE HIT
:*
:* VERIFY THAT 'VALID' INPUT TO CACHE HIT NAND GATE WILL INHIBIT NAND
:* GATE FROM INDICATING A CACHE HIT.
:*
:* PROCEDURE: CREATE A CONDITION WHERE ONLY VALID INPUT ON
:* CACHE HIT NAND GATE INHIBITS NAND GATE:
:* 1.UPDATE CACHE LOCATION 0000
:* 2.CAUSE INVALIDATION BY A WRITE HIT
:* IN BYPASS MODE
:* 3.CAUSE READ HIT
:*
:* VERIFY THAT OUTPUT OF NAND GATE
:* WILL READ AS A 1 THRU CMR<8>.
:*
:* CONDITIONS: INPUTS CACHE HIT NAND GATE:
:* COMPARE 1 =1
:* COMPARE 2 =1
:* COMPARE 3 =1
:* VALID =0
:* TAG PAR. ERR -1
:* HI BYTE PE =1
:* LO BYTE PE =1
:* MISS HI -1
:* MISS LO =1
:* BYPASS/WRITE =1
:* FAULT =1
:*****
```

```
040736
040736 000004
040740 040750
040742 070012
040744 070054
040746 070064
040750 012737 001015 177746 40$: MOV #OFF,CCR
040756 004437 002452 JSR R4,RELCTH
040762 041100 .WORD 10$+2
:SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
:TEST START LOCATION
:LOOP ON ERROR START LOCATION
:SCOPE SYNC. 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
```

```
6592 040764 005000 CLR R0 ;CLEAR R0
6593 040766 005037 060000 CLR 60000 ;ALL 0'S TO MAIN MEMORY LOC. 60000
6594 040772 012701 060000 MOV #60000,R1 ;ADDRESS 60000 TO R1
6595 040776 112737 000002 177750 1$: MOVB #HODO,CMR ;HODO ALLOWS READ HITS TO BE CACHED,CACHE UPDATES,AND
6596 ;CLOCKING OF OUTPUT OF CACHE HIT NAND
6597 ;GATE INTO CMR ONLY DURING THE DESTINATION
6598 ;ACCESS OF AN INSTRUCTION.
6599 041004 012737 000011 177746 MOV #11,CCR ;NO UCB SO AS TO WRITE CACHE STORES
6600 ;ENABLE LO CACHE
6601 041012 005737 040000 TST 40000
6602 041016 005737 060000 TST 60000 ;READ UPDATE; ASSURE CORRECT PARITY IS WRITTEN
6603 ;FOR CACHE LOCATION 0000
6604 041022 052737 001000 177746 BIS #UCB,CCR ;SET UCB SO AS TO INVALIDATE CACHE LOCATIONS
6605 ;DURING WRITE HIT
6606 041030 010011 MOV R0,(R1) ;CAUSE WRITE HIT TO LOCATION 60000;
6607 ;UCB CAUSES CACHE LOC. 0000 TO BE INVALIDATED
6608 041032 042737 001000 177746 BIC #UCB,CCR ;CLEAR UCB
```

6609	041040	005737	060000	20\$:	TST	60000		:READ UPDATE CAUSED BY VALID STORE
6610								:INVALIDATED; ALL INPUTS TO CACHE HIT NAND GATE
6611								:ARE 1 EXCEPT VALID.CLOCK STATUS OF NAND GATE
6612								:OUTPUT TO CMR<8>
6613	041044	013702	177750		MOV	CMR,R2		:SAVE CMR CONTENTS
6614	041050	000240		25\$:	NOP			:INSTRUCTION 'JMP 1\$' PLACED HERE
	041052	00C240			NOP			:FOR LOOP ON ERROR
6615	041054	012737	001015	177746	MOV	#OFF,CCR		:DISABLE CACHE
6616	041062	032702	000400		BIT	#HIT,R2		:WAS CACHE HIT SIGNAL A 1
6617	041066	001003			BNE	10\$:PASS
6618	041070	104413			ERROR			:ERROR
								:-----
	041072	041070			.WORD	.-2		:CACHE HIT TESTS
6619								:READING OUTPUT OF CACHE HIT NAND GATE
6620								:THRU CMR<8> DID NOT RESULT IN A 1
6621								
6622	041074	000000			.WORD	0		:END OF TEST
6623	041076	000240		10\$:	NOP			:INCREMENT TEST COUNTER
	041100	005237	001472		INC	\$TESTN		

6644

```

SBTTL TEST # 214 - 'COMPARE 1' INPUT STOPS GATE FROM IND. HIT
*****
*TEST 214 'COMPARE 1' INPUT STOPS GATE FROM IND. HIT
* VERIFY THAT 'COMPARE 1' INPUT TO CACHE HIT NAND GATE CAN INHIBIT
* NAND GATE FROM INDICATING A CACHE HIT.
* PROCEDURE: CREATE A READ UPDATE TO LOW CACHE CAUSED BY ONLY
* BIT 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<B>.
* 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
    
```

```

041104
041104 000004
041106 041116
041110 070000
041112 070066
041114 070106
041116 012737 001015 177746
041124 004437 002452
041130 041312
    
```

```

*ST214:
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
MOV #OFF,CCR ;DISABLE CACHE
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

```

6645 041132 012737 070100 000004
6646 041140 012737 000340 000006
6647 041146 012737 010000 172350
6648 041154 012700 100000
6649
6650
6651
6652 041160 012737 000001 177572
6653 041166 012737 000020 172516
6654 041174 112737 000002 177750
6655
6656
6657
6658 041202 012737 000011 177746
6659
6660 041210 005737 040000
6661 041214 005737 000000
6662
6663 041220 005710
6664
6665
    
```

```

1$: MOV #3$-1$+70000,4 ;SETUP FOR POTENTIAL TRAP
MOV #340,6
MOV #10000,KPAR4 ;MAP PAGE 4 FOR 128K-132K ADDRESSING
MOV #100000,R0 ;LOAD VIRTUAL ADDRESS IN R0.WHEN MEMORY
;MANAGEMENT IS ENABLED,PAGE 4 WILL
;BE SELECTED AND ADDRESS 1000000
;WILL BE ACCESSED.
MOV #1,SR0 ;ENABLE MEM. MNGMENT.
MOV #20,SR3 ;ENABLE 22-BIT MAPPING
MOV #HODO,CMR ;HODO ALLOWS READ HITS,UPDATES, AND
;CLOCKING OF OUTPUT OF CACHE HIT NAND
;GATE INTO CMR ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
MOV #11,CCR ;NO UCB SO AS TO WRITE CACHE STORES
;ENABLE LOW CACHE
;
TST 40000
TST 0
;READ UPDATE; ASSURE ALL 0'S IN TAG STORE
;LOCATION 0000,AND CORRECT PARITY IS WRITTEN.
20$: TST (R0) ;READ UPDATE TO CACHE LOCATION 0000
;CAUSED BY BIT 18 ON CACHE ADDRESS LINE
;DIFFERENT FROM TAG STORE BIT 18.
    
```

```

6666                                     ; CLOCK STATUS OF NAND GATE
6667                                     ; OUTPUT TO CMR<8>
6668 041222 000240                       NOP
6669 041224 013701 177750                 MOV    CMR,R1           ; SAVE CMR CONTENTS
6670 041230 000403                       BR     25$
6671 041232 013701 177750                 3$:   MOV    CMR,R1           ; IF TRAP OCCURS WILL RETURN HERE
6672 041236 022626                       CMP    (SP)+,(SP)+     ; ADJUST STACK AND VECTORS
6673 041240 000240                       25$:  NOP
        041242 000240                       NOP
6674 041244 005037 177572                 CLR    SR0             ; DISABLE MEM. MNGMENT.
6675 041250 005037 172516                 CLR    SR3             ; DISABLE 22 BIT MAPPING
6676 041254 012737 001015 177746         MOV    #OFF,CCR        ; DISABLE CACHE
6677 041262 012737 000006 000004         MOV    #6,4           ; RESTORE VECTORS
6678 041270 005037 000006                 CLR    6
6679 041274 032701 000400                 BIT    #HIT,R1        ; WAS CACHE HIT SIGNAL A 1
6680 041300 001003                       BNE   10$
6681 041302 104413                       ERROR
        041304 041302                       .WORD  -2
6682                                     ; CACHE HIT TESTS
6683                                     ; READING OUTPUT OF CACHE HIT NAND GATE
6684                                     ; THRU CMR<8> DID NOT RESULT IN A 1
6685 041306 000000                       .WORD  0
6686 041310 000240                       10$:  NOP
        041312 C05237 001472                 INC    $TESTN         ; END OF TEST
        ; INCREMENT TEST COUNTER
    
```

6707

```

.SBTTL TEST # 215 - 'COMPARE 2' INPUT STOPS NAND FROM IND. HIT
*****
:TEST 215 'COMPARE 2' INPUT STOPS NAND FROM IND. HIT
:VERIFY THAT 'COMPARE 2' INPUT TO CACHE HIT NAND GATE CAN INHIBIT
:  NAND GATE FROM INDICATING A CACHE HIT.
:PROCEDURE: CREATE A READ UPDATE TO LOW CACHE CAUSED BY ONLY
:  BIT 14 ON CACHE ADDRESS LINE BEING DIFFERENT
:  FROM BIT 14 IN TAG STORE. VERIFY THAT OUTPUT OF CACHE HIT
:  NAND GATE WILL READ AS A 1 THRU CMR<8>.
:CONDITIONS: INPUTS CACHE HIT NAND GATE:
:  COMPARE 1 =1
:  COMPARE 2 =0
:  COMPARE 3 =-1
:  VALID =1
:  TAG PAR. ERR =1
:  HI BYTE PE =1
:  LO BYTE PE =1
:  MISS HI =1
:  MISS LO =1
:  BYPASS/WRITE =1
:  FAULT =1
*****
  
```

```

041316
041316 000004          SPCOND          :SCOPE CONDITIONS:GO SET UP FOR LOOP ON
:ERROR/LOOP ON TEST
041320 041330          .WORD 40$          :TEST START LOCATION
041322 070000          .WORD 1$-40$+67764      :LOOP ON ERROR START LOCATION
041324 070024          .WORD 20$-40$+67764    :SCOPE SYNC. LOCATION
041326 070034          .WORD 25$-40$+67764    :LOOP ON ERROR END LOCATION
041330 012737 001015 177746 40$: MOV #OFF,CCR :DISABLE CACHE
041336 004437 002452          JSR R4,RELCTH          :LOCATE TEST CODE TO HIGH CACHE SPACE
041342 041430          .WORD 10$+2          :ADDRESS OF START OF NEXT TEST
  
```

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

```

6708 041344 112737 000002 177750 1$: MOV #HODO,CMR :HODO ALLOWS READ HITS,UPDATES, AND
6709 :CLOCKING OF OUTPUT OF CACHE HIT NAND
6710 :GATE INTO CMR ONLY DURING THE DESTINATION
6711 :ACCESS OF AN INSTRUCTION.
6712 041352 012737 000011 177746 MOV #11,CCR :NO UCB SO AS TO WRITE CACHE STORES
6713 :ENABLE LOW CACHE
6714 041360 005737 040000 TST 40000 :
6715 041364 005737 000000 TST 0 :READ UPDATE; ASSURE ALL 0'S IN TAG STORE
6716 :LOCATION 0000,AND CORRECT PARITY IS WRITTEN.
6717 041370 005737 040000 20$: TST 40000 :READ UPDATE TO CACHE LOCATION 0000
6718 :CAUSED BY BIT 14 ON CACHE ADDR=SS LINE
6719 :DIFFERENT FROM TAG STORE BIT 14.
6720 :CLOCK STATUS OF NAND GATE
6721 :OUTPUT TO CMR<8>
6722 041374 013701 177750 MOV CMR,R1 :SAVE CMR CONTENTS
6723 041400 000240 25$: NOP :INSTRUCTION 'JMP 1$' PLACED HERE
6724 041402 000240 NOP :FOR LOOP ON ERROR
6724 041404 012737 001015 177746 MOV #OFF,CCR :DISABLE CACHE
6725 041412 032701 000400 BIT #HIT,R1 :WAS CACHE HIT SIGNAL A 1
6726 041416 001003 BNE 10$ :PASS
6727 041420 104413 ERROR :ERROR
  
```

	041422	041420	
6728			
6729			
6730			
6731	041424	000000	
6732	041426	000240	
	041430	005237	001472

	.WORD	.-2
10\$:	.WORD	0
	NOP	
	INC	\$TESTN

```

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

```

'COMPARE 3' INPUT STOPS GATE FROM IND. HIT

.SBTTL TEST # 216 - 'COMPARE 3' INPUT STOPS GATE FROM IND. HIT

.....

TEST 216 'COMPARE 3' INPUT STOPS GATE FROM IND. HIT

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

* PROCEDURE: CREATE A READ UPDATE TO LO CACHE CAUSED BY ONLY BIT 13 ON CACHE ADDRESS LINE BEING DIFFERENT FROM BIT 13 IN TAG STORE.

* VERIFY THAT THE OUTPUT OF CACH HIT NAND GATE WILL READ AS A 1 THRU CMR<8>.

* CONDITIONS: INPUTS CACHE HIT NAND GATE:

* COMPARE 1 =-1
* COMPARE 2 =1
* COMPARE 3 =0
* VALID =1
* TAG PAR. ERR =-1
* HI BYTE PE =1
* LO BYTE PE =-1
* MISS HI =1
* MISS LO =1
* BYPASS/WRITE =-1
* FAULT =1

.....

041434
041434 000004

TST216: SPCOND ;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
40\$: MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
.WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

041436 041446
041440 070000
041442 070024
041444 070034
041446 012737 001015 177746
041454 004437 002452
041460 041546

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

6755 041462 112737 000002 177750 1\$: MOVB #HODO,CMR
6756
6757
6758
6759 041470 012737 000011 177746 MOV #11,CCR
6760
6761 041476 005737 020000 TST 20000
6762 041502 005737 000000 TST 0
6763
6764
6765 041506 005737 020000 20\$: TST 20000
6766
6767
6768
6769
6770 041512 013701 177750 MOV CMR,R1
6771 041516 000240 25\$: NOP
041520 000240 NOP
6772 041522 012737 001015 177746 MOV #OFF,CCR
6773 041530 032701 000400 BIT #HIT,R1

;HODO ALLOWS READ HITS,UPDATES, AND
;CLOCKING OF OUTPUT OF CACHE HIT NAND
;GATE INTO CMR ONLY DURING THE DESTINATION
;ACCESS OF AN INSTRUCTION.
;NO UCB SO AS TO UPDATE CACHE STORES
;ENABLE LOW CACHE
;
;READ UPDATE; ASSURE ALL 0'S IN TAG
;STORE LOCATION 0000 AND CORRECT PARITY
;IS WRITTEN.
;READ UPDATE TO CACHE LOC. 0000 CAUSED BY
;BIT 13 ON CACHE ADDRESS LINES BEING
;DIFFERENT FROM BIT 13 IN TAG STORE.
;CLOCK STATUS OF CACHE HIT NAND GATE
;OUTPUT TO CMR<8>
;SAVE CMR CONTENTS
;INSTRUCTION 'JMP 1\$' PLACED HERE
;FOR LOOP ON ERROR
;DISABLE CACHE
;WAS CACHE HIT SIGNAL A 1

TEST # 216 - 'COMPARE 3' INPUT STOPS GATE FROM IND. HIT

6774 041534 001003
6775 041536 104413

BNE 10\$
ERROR

:PASS
:ERROR
:-----

041540 041536

.WORD .-2

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

6776
6777
6778

6779 041542 000000
6780 041544 000240
041546 005237 001472

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

:END OF TEST
:INCREMENT TEST COUNTER

6787

```

.SBTTL TEST # 217 - CACHE READ HIT RESULTS IN PROPER OUTPUT
:*****
:*TEST 217 CACHE READ HIT RESULTS IN PROPER OUTPUT
:* VERIFY THAT A CACHE READ HIT WILL RESULT IN DATA BEING READ
:* FROM CACHE DATA STORE, ASSURING THAT THE CACHE HAS ISSUED A
:* A CPU CLOCK RESTART SIGNAL. ASSURE THAT ALL 0'S CAN BE CACHED
:* OUT OF CACHE DATA STORE.
:*****

```

041552
041552 000004

041554 041564
041556 070000
041560 000000
041562 070106
041564 012737 001015 177746
041572 004437 002452
041576 042030

```

TST217:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
;DISABLE CACHE
;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST

```

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

6788 041600 012701 040000
6789 041604 012702 060000
6790 041610 012737 070076 000014
6791 041616 012737 000340 000016
6792 041624 005037 002070
6793 041630 005037 002072
6794 041634 012706 060002
6795 041640 005037 060000
6796
6797 041644 012737 000340 177776
6798 041652 112737 000002 177750
6799
6800
6801
6802
6803 041660 012737 000011 177746
6804
6805 041666 000257
6806 041670 005711
6807 041672 005712
6808
6809
6810 041674 000003
6811
6812
6813
6814
6815
6816
6817
6818
6819 041676 042737 000002 177750 3\$:
6820 041704 011200
6821
6822

```

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
MOV #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
TST (R1) ;CLEAR ALL CONDITION CODES
TST (R2)
;CACHE READ UPDATE. WRITE ALL 0'S FROM
;MAIN MEMORY LOCATION TO CACHE DATA STORE
;LOCATION 0000.
BPT ;BREAKPOINT TRAP. DUE TO A TRAP,THE PSW
;WILL BE WRITTEN TO THE STACK, WHICH NOW
;POINTS TO ADDRESS 60000.THE TRAP INSTRUCTION
;IS A NON-DESTINATION ACCESS INSTR.
;SINCE HODO IS BEING USED, A CACHE UPDATE
;WILL BE INHIBITED. MAIN MEMORY
;ADDRESS 60000 WILL CONTAIN PSW DATA OF 344,AND
;THE LOCATION IN CACHE CORRESPONDING TO ADDRESS
;60000 WILL BE LEFT WITH ALL 0'S DATA.
BIC #HODO,CMR ;TRAP TO HERE;DISABLE HODO
MOV (R2),RO ;WHEN THIS INSTRUCTION READS
;ADDRESS 60000
;A CACHE READ HIT SHOULD RESULT AND A CPU CLOCK

```


6863

.SBTTL TEST # 220 - FLOATING 1 CAN BE CACHED FROM DATA STORE
 :*****
 :*TEST 220 FLOATING 1 CAN BE CACHED FROM DATA STORE
 :* VERIFY THAT A FLOATING 1 ACROSS 0'S DATA PATTERN CAN BE CACHED
 :* FROM CACHE DATA STORE.
 :*****

042034
 042034 000004

042036 042046
 042040 070016
 042042 000000
 042044 070040
 042046 012737 001015 177746
 042054 004437 002452
 042060 042176

TST220:
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 ;ERROR/LOOP ON TEST
 .WORD 40\$;TEST START LOCATION
 .WORD 1\$-40\$+67764 ;LOOP ON ERROR START LOCATION
 .WORD 0 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
 .WORD 25\$-40\$+67764 ;LOOP ON ERROR END LOCATION
 MOV #OFF,CCR ;DISABLE CACHE
 JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
 .WORD 10\$+2 ;ADDRESS OF START OF NEXT TEST

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

6864 042062 012701 040000
 6865 042066 012702 060000
 6866 042072 012737 000001 050516
 6867 042100 013737 050516 060000 1\$:
 6868
 6869 042106 012737 000011 177746
 6870
 6871 042114 005711
 6872 042116 005712
 6873
 6874
 6875 042120 011200
 6876
 6877
 6878
 6879
 6880 042122 000240 25\$:
 042124 000240
 6881 042126 012737 001015 177746
 6882 042134 020037 050516
 6883 042140 001412
 6884 042142 010037 050520
 6885 042146 013737 050516 050504
 6886 042154 104413
 042156 042154
 6887
 6888
 6889
 6890
 6891 042160 050504
 6892
 6893 042162 050520
 6894 042164 000000
 6895 042166 006337 050516 9\$:
 6896 042172 103342
 6897

MOV #40000,R1 ;ADDRESS 40000 TO R1
 MOV #60000,R2 ;ADDRESS 60000 TO R2
 MOV #1,FLTPAT ;1ST FLOATING 1 PATTERN: 000001
 MOV FLTPAT,60000 ;WRITE FLOATING 1 PATTERN TO MAIN MEMORY
 ;LOCATION 60000
 MOV #11,CCR ;NO BYPASS TO ALLOW WRITES TO CACHE STORES.
 ;ENABLE LOW CACHE
 TST (R1)
 TST (R2) ;CACHE READ UPDATE. WRITE FLOATING 1 PATTERN FROM
 ;MAIN MEMORY LOCATION TO CACHE DATA STORE
 ;LOCATION 0000.
 MOV (R2),R0 ;WHEN THIS INSTRUCTION READS
 ;ADDRESS 60000,A CACHE READ HIT SHOULD RESULT
 ;AND A CPU CLOCK RESTART SIGNAL SHOULD BE ISSUED.
 ;THE CPU SHOULD READ FLOATING 1 PATTERN FROM CACHE DATA STORE
 ;RATHER THAN MAIN MEMORY.
 ;INSTRUCTION 'JMP 1\$' PLACED HERE
 ;FOR LOOP ON ERROR
 MOV #OFF,CCR ;DISABLE CACHE
 CMP R0,FLTPAT ;WAS THE CORRECT FLOATING 1 PATTERN RECEIVED
 BEQ 9\$;PASS
 MOV R0,RECDAT ;GET DATA RECEIVED
 MOV FLTPAT,EXDAT6 ;GET EXPECTED DATA
 ERROR ;ERROR
 ;-----
 .WORD -2
 ;CPU CLOCK RESTART-CACHED DATA TESTS
 ;CREATING A READ HIT BY READING ADDRESS 60000
 ;RESULTED IN INCORRECT FLOATING 1 PATTERN
 ;BEING CACHED FROM CACHE DATA STORE
 ;PRINT FLOATING 1 PATTERN EXPECTED FROM THE
 ;READ HIT TO ADDRESS 60000
 ;PRINT DATA RECEIVED FROM READ HIT TO ADDRESS 60000
 EXDAT6
 RECDAT
 .WORD 0
 ASL FLTPAT ;NEXT FLOATING 1 PATTERN
 BCC 1\$;IF FLOATING 1 PATTERN 100000 HAS NOT BEEN
 ;DONE ,CONTINUE.

6898 042174 000240 10\$: NOP ;END OF TEST
 042176 005237 001472 INC \$TESTN ;INCREMENT TEST COUNTER

6903

```

.SBTTL TEST # 221 - DMA WRITE HITS STOPS ALL CACHE STORE LOCS
:*****
:*TEST 221 DMA WRITE HITS STOPS ALL CACHE STORE LOCS
:* VERIFY THAT DMA WRITE HITS WILL INVALIDATE ALL OF LOW CACHE VALID
:* STORE LOCATIONS.
:*****
    
```

```

042202
042202 000004

042204 042214
042206 070270
042210 000000
042212 070410
042214 012737
042222 004437
042226 042726
    
```

```

TST221:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
.WORD 40$ ;LOOP ON ERROR START LOCATION
.WORD 1$-40$+67764 ;SCOPE SYNC, NOT IMPLEMENTED FOR THIS TEST
.WORD 0 ;LOOP ON ERROR END LOCATION
.WORD 25$-40$+67764 ;DISABLE CACHE
MOV #OFF,CCR ;LOCATE TEST CODE TO HIGH CACHE SPACE
JSR R4,RELCTH ;ADDRESS OF START OF NEXT TEST
.WORD 10$+2
    
```

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

```

6904 042230 032737 020000 177746 BIT #VSIU,CCR ;IS SET A BEING USED?
6905 042236 001407 BEQ 7$ ;YES
6906 042240 052737 000400 177746 BIS #FC,CCR ;NO; FLUSH CACHE FOR SET A
6907 042246 032737 010000 177746 500$: BIT #VCIP,CCR ;WAIT TILL FLUSH COMPLETE
6908 042254 001374 BNE 500$
6909 042256 012705 060000 7$: MOV #60000,R5 ;;ADDRESS 60000 INTO R5
6910 042262 012703 040000 MOV #40000,R3 ;ADDRESS 40000 INTO R3
6911 042266 132737 000200 001507 BITB #APTSIZE,$ENVM ;WILL APT SIZE?
6912 042274 001405 BEQ 3$ ;NO,GO AUTOSIZE
6913 042276 032737 010000 001512 BIT #10000,$USWR ;DOES APT SAY TO PERFORM TEST
6914 042304 001426 BEQ 11$ ;APT SAYS DO NOT PERFORM TEST
6915 042306 000504 BR 1$ ;APT SAYS DO TEST
6916 042310 012737 070116 000004 3$: MOV #5$-40$+67764,4 ;SETUP FOR TRAP
6917 042316 012737 000340 000006 MOV #340,6
6918 042324 005737 170006 TST BECR1 ;ACCESS UNIBUS EXERCISER
6919 042330 000240 NOP
6920 042332 012737 000006 000004 MOV #6,4 ;RESTORE VECTORS
6921 042340 005037 000006 CLR 6
6922 042344 000465 BR 1$ ;UNIBUS EXERCISER IS PRESENT;PROCEED WITH TEST
6923 042346 022626 5$: CMP (S. )+,(SP)+ ;TRAP RETURN;EXERCISER NOT PRESENT
6924 042350 012737 000006 000004 MOV #6,4 ;RESTORE VECTORS
6925 042356 005037 000006 CLR 6
6926 042362 005737 001474 11$: TST $PASS ;IS THIS FIRST PASS?
6927 042366 001156 BNE 10$ ;SKIP MESSAGE;SKIP TEST
6928 042370 023737 000042 000046 CMP 42,46 ;IS THIS ACTION QV OR AUTO ACCEPT?
6929 042376 001552 BEQ 10$ ;YES SKIP TYPEOUT
6930 042400 104401 042406 TYPE ,65$ ;;TYPE ASCIZ STRING
042404 000402 BR 64$ ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <CRLF><CRLF>
64$:
042412
6931 042412 104401 042420 TYPE ,67$ ;;TYPE ASCIZ STRING
042416 000432 BR 66$ ;;GET OVER THE ASCIZ
;;67$: .ASCIZ /UNIBUS EXERCISER NOT USED- DMA TESTS NOT PERFORMED/
66$:
042504
6932 042504 104401 042512 TYPE ,69$ ;;TYPE ASCIZ STRING
042510 000402 BR 68$ ;;GET OVER THE ASCIZ
;;69$: .ASCIZ <CRLF><CRLF>
    
```

```

68$:
6933 042516 000502          BR      10$
6934 042520 012737 000015 177746 1$:  MOV    #15,CCR      ;NO UCB SO AS TO WRITE ENABLE VALID STORE
6935 042526 112737 000002 177750  MOVB   #HODO,CMR    ;HODO ALLOWS UPDATES ONLY DURING THE
6936                                ;DESTINATION ACCESS OF AN INSTRUCTION
6937 042534 005723          2$:  TST    (R3)+
6938 042536 005725          TST    (R5)+      ;UPDATE ALL LOW CACHE LOCATIONS MAKING
6939                                ;ALL VALID STORE LOCATIONS =1
6940 042540 022705 070000          CMP    #70000,R5   ;COMPLETE?
6941 042544 001373          BNE    2$         ;NO
6942 042546 042737 000002 177750  BIC    #HODO,CMR   ;CLEAR HODO SO VALID STORE CAN BE WRITTEN
6943                                ;BY UNIBUS EXERCISER.
6944 042554 012705 060000          MOV    #60000,R5   ;ADDRESS 60000 INTO R5
6945 042560 012737 060000 170004  MOV    #60000,BEBA ;SETUP UNIBUS EXERCISER
6946                                ;ADRESS
6947 042566 012737 174000 170002  MOV    #-4000,BECC ;TRANSFER COUNT
6948 042574 012737 177777 170000  MOV    #177777,BEDA ;DATA FOR WRITE XFER
6949 042602 012737 000000 170016  MOV    #0,BECCR2   ;SETUP CONTROL REGISTER 2
6950 042610 012737 003045 170006  MOV    #3045,BECCR1 ;SETUP CONTROL REGISTER 1;START XFER
6951 042616 105737 170006          4$:  TSTB   BECCR1     ;WAIT FOR EXERCISER TO COMPLETE
6952 042622 100375          BPL    4$
6953 042624 052737 000002 177750  6$:  BIS    #HODO,CMR   ;IMPLEMENT HODO. ALLOWS VALID STORE
6954                                ;DATA TO BE WRITTEN TO CMR<12> ONLY
6955                                ;DURING THE DESTINATION MEMORY ACCESS
6956                                ;OF AN INSTRUCTION.
6957 042632 005715          TST    (R5)       ;READ LOW CACHE ADDRESS
6958                                ;LOCATION SPECIFIED BY R5'S BITS 12 TO 1.
6959                                ;WRITE VALID STORE DATA INTO CMR<12>
6960                                ;FROM VALID STORE ADDRESS LOCATION
6961                                ;JUST READ.
6962 042634 013701 177750          MOV    CMR,R1     ;SAVE CMR DATA
6963 042640 000240          25$:  NOP
6963 042642 000240          NOP
6964 042644 012737 001015 177746  MOV    #OFF,CCR   ;INSTRUCTION 'JMP 1$' PLACED HERE
6965 042652 105037 177750          CLRB   CMR        ;FOR LOOP ON ERROR
6966 042656 032701 010000          BIT    #VLD,R1   ;DISABLE CACHE
6967 042662 001411          BEQ    9$         ;DISABLE MAINTENANCE
6968 042664 010537 050506          MOV    R5,CA121  ;CMR<12> SHOULD BE 0.
6969                                ;PASS
6970 042670 006237 050506          ASR    CA121     ;SAVE VALID STORE ADDRESS LOCATION
6971 042674 104413          ERROR  CA121    ;USED: CA<12:1>
6972                                ;PREPARE CA121 FOR TYPEOUT
6973                                ;ERROR
6974                                ;-----
6975                                ;DMA TESTS
6976                                ;READING VALID STORE DATA
6977 042700 050506          CA121
6978                                ;THRU CMR<12> DID NOT RESULT IN 0.
6979                                ;THIS INDICATES THAT VALID STORE WAS
6980                                ;NOT INVALIDATED DUE TO DMA WRITE HIT.
6981                                ;PRINT VALID STORE ADDRESS LOCATION
6982                                ;USED: CA<12:1>.
6983                                ;IF ERROR END TEST
6984                                ;NEXT VALID STORE LOCATION
6985 042722 001340          9$:  .WORD 0
6985                                ;HAVE ALL LOW CACHE ADDRESS LOCATIONS
6985                                ;BEEN DONE?
6985                                ;NO

```

6986 042724 000240
042726 005237 C01472

108: NOP
IN STEIN

:END OF TEST
:INCREMENT TEST COUNTER

6991

.SBTTL TEST # 222 - DMA WRITE CAUSES TIMEOUT & CCR REG NOT ALTERED
 :*****
 :*TEST 222 DMA WRITE CAUSES TIMEOUT & CCR REG NOT ALTERED
 :* VERIFY THAT A DMA WRITE TO CCR REGISTER WILL RESULT IN A TIMEOUT
 :* AND THAT THE CCR REGISTER WILL NOT BE ALTERED
 :*****

042732
 042732 000004

042734 042744
 042736 070134
 042740 000000
 042742 070276
 042744 012737 001015 177746
 042752 004437 002452
 042756 043322

TST222:
 SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
 .WORD 40\$;ERROR/LOOP ON TEST
 .WORD 1\$-40\$+67764 ;TEST START LOCATION
 .WORD 0 ;LOOP ON ERROR START LOCATION
 .WORD 25\$-40\$+67764 ;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
 MOV #OFF,CCR ;LOOP ON ERROR END LOCATION
 JSR R4,RELCTH ;DISABLE CACHE
 .WORD 10\$+2 ;LOCATE TEST CODE TO HIGH CACHE SPACE
 ;ADDRESS OF START OF NEXT TEST

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

6992	042760	032737	020000	177746	BIT	#VSIU,CCR	;IS SET A BEING USED?
6993	042766	001407			BEQ	7\$;YES
6994	042770	052737	000400	177746	BIS	#FC,CCR	;NO; FLUSH CACHE FOR SET A
6995	042776	032737	010000	177746	500\$: BIT	#VCIP,CCR	;WAIT TILL FLUSH COMPLETE
6996	043004	001374			BNE	500\$	
6997	043006	012705	060000		7\$: MOV	#60000,R5	::ADDRESS 60000 INTO R5
6998	043012	012703	040000		MOV	#40000,R3	;ADDRESS 40000 INTO R3
6999	043016	132737	000200	001507	BITB	#APTSIZE,\$ENVM	;WILL APT SIZE?
7000	043024	001405			BEQ	3\$;NO,GO AUTOSIZE
7001	043026	032737	010000	001512	BIT	#10000,\$USWR	;DOES APT SAY TO PERFORM TEST
7002	043034	001426			BEQ	11\$;APT SAYS DO NOT PERFORM TEST
7003	043036	000426			BR	1\$;APT SAYS DO TEST
7004	043040	012737	070116	000004	3\$: MOV	#5\$-40\$+67764,4	;SETUP FOR TRAP
7005	043046	012737	000340	000006	MOV	#340,6	
7006	043054	005737	170006		TST	BECR1	;ACCESS UNIBUS EXERCISER
7007	043060	000240			NOP		
7008	043062	012737	000006	000004	MOV	#6,4	;RESTORE VECTORS
7009	043070	005037	000006		CLR	6	
7010	043074	000407			BR	1\$;UNIBUS EXERCISER IS PRESENT;PROCEED WITH TEST
7011	043076	022626			5\$: CMP	(SP)+,(SP)+	;TRAP RETURN;EXERCISER NOT PRESENT
7012	043100	012737	000006	000004	MOV	#6,4	;RESTORE VECTORS
7013	043106	005037	000006		CLR	6	
7014	043112	000502			11\$: BR	10\$;SKIP TEST
7015	043114	012737	000015	177746	1\$: MOV	#15,CCR	;CACHE OFF-DISABLE INTERRUPT
7016	043122	012737	070256	000510	MOV	#6\$-40\$+67764,510	;SETUP RETURN ADDRESS FOR
7017							;A UNIBUS EXER. TRAP
7018	043130	012737	000340	000512	MOV	#340,512	
7019	043136	012737	177746	170004	MOV	#177746,BEBA	;SETUP UNIBUS EXERCISER ADDRESS
7020	043144	012737	177777	170002	MOV	#-1,BECC	;TRANSFER COUNT
7021	043152	012737	001015	170000	MOV	#1015,BEDA	;DATA FOR WRITE XFER
7022	043160	012737	000003	170016	MOV	#3,BECC2	;SETUP CONTROL REGISTER 2
7023	043166	012737	003045	170006	MOV	#3045,BECC1	;SETUP CONTROL REGISTER 1;START XFER
7024	043174	105737	170006		4\$: TSTB	BECC1	;WAIT FOR EXERCISER TO COMPLETE
7025	043200	100375			BPL	4\$	
7026	043202	012737	001000	002062	MOV	#1000,LOOP	;GIVE ENOUGH TIME FOR TIMEOUT TO OCCUR
7027	043210	005337	002062		2\$: DEC	LOOP	
7028	043214	001375			BNE	2\$	

7061

.SBTTL TEST # 223 - ALL 6 HIT REG BITS READ 0 DUE TO 6 READ MISSES

:TEST 223 ALL 6 HIT REG BITS READ 0 DUE TO 6 READ MISSES
:* CHECK THAT ALL SIX HIT REGISTER BITS CAN READ 0 DUE TO SIX
:* READ MISSES
:*****

TST223:

043326
043326 000004

SCPCND

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

043330 043340
043332 070010
043334 000000
043336 070046
043340 012737
043346 004437
043352 043466

001015 177746 40\$:
002452

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

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

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

7062 043354 012700 040000
7063 043360 012701 060000
7064 043364 112737 000002 177750 1\$:
7065
7066
7067 043372 012737 000015 177746
7068 043400 005710
7069 043402 005711
7070 043404 005710
7071 043406 005711
7072 043410 005710
7073 043412 005711
7074 043414 005710
7075 043416 013702 177752
7076 043422 000240 25\$:
043424 000240
7077 043426 012737 001015 177746
7078 043434 105037 177750
7079 043440 042702 177700
7080 043444 005702
7081 043446 001406
7082 043450 010237 050524
7083 043454 104413

MOV #40000,R0
MOV #60000,R1
MOVB #HODO,CMR

MOV #15,CCR
TST 'R0'
TST (R1)
TST (R0)
TST (R1)
TST (R0)
TST (R1)
TST (R0)
TST (R1)
TST (R0)
MOV CHR,R2
NOP
NOP
MOV #OFF,CCR
CLRB CMR
BIC #177700,R2
TST R2
BEQ 10\$
MOV R2,CHR50
ERROR

:ADDR. 40000 TO R0
:ADDR 60000 TO R1
:HODO ALLOWS HIT REGISTER TO BE CLOCKED
:ONLY DURING THE DESTINATION ACCESS
:OF AN INSTRUCTION.
:NO UCB SO AS TO WRITE CACHE STORES

:READ MISS
:READ MISS
:READ MISS
:READ MISS
:READ MISS
:READ MISS
:READ MISS
:SAVE CHR CONTENTS
:INSTRUCTION 'JMP 1\$' PLACED HERE
:FOR LOOP ON ERROR
:DISABLE CACHE
:DISABLE MAINTENANCE MODE
:PREPARE R2 FOR CHECK
:CHR<5:0> SHOULD HAVE BEEN ALL 0'S
:PASS
:PREPARE FOR ERROR REPORT
:ERROR
:-----

043456 043454
7084
7085
7086 043460 050524
7087 043462 000000
7088 043464 000240 10\$:
043466 005237 001472

.WORD -2

CHR50
.WORD 0
NOP
INC \$TESTN

:CHR<5:0> DID NOT INDICATE ALL 0'S
:DUE TO SIX READ MISSES
:PRINT CHR<5:0> RECEIVED

:END OF TEST
:INCREMENT TEST COUNTER

7093

```
.SBTTL TEST # 224 - BIT 05 OF HIT REG CONTAINS 1 WITH 1 HIT, 5 MISSES
:*****
:TEST 224 BIT 05 OF HIT REG CONTAINS 1 WITH 1 HIT, 5 MISSES
:* VERIFY THAT BIT 05 OF CACHE HIT REGISTER CAN CONTAIN A 1 DUE TO
:* ONE READ HIT FOLLOWED BY FIVE READ MISSES
:*****
```

```
043472
043472 000004

043474 043504
043476 070010
043500 000000
043502 070046
043504 012737 001015 177746
043512 004437 002452
043516 043634
```

```
TST224:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION
MOV #OFF,CCR ;DISABLE CACHE
JSR R4,RELCTH ;LOCATE TEST CODE TO HIGH CACHE SPACE
;ADDRESS OF START OF NEXT TEST
```

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

```
7094 043520 012700 040000
7095 043524 012701 060000
7096 043530 112737 000002 177750
7097
7098
7099 043536 012737 000015 177746
7100 043544 005710
7101 043546 005710
7102 043550 005711
7103 043552 005710
7104 043554 005711
7105 043556 005710
7106 043560 005711
7107 043562 013702 177752
7108 043566 000240
043570 000240
7109 043572 012737 001015 177746
7110 043600 105037 177750
7111 043604 042702 177700
7112 043610 022702 000040
7113 043614 001406
7114 043616 010237 050524
7115 043622 104413

043624 043622

7116
7117
7118 043626 050524
7119 043630 000000
7120 043632 000240
043634 005237 001472
```

```
MOV #40000,R0 ;ADDR. 40000 TO R0
MOV #60000,R1 ;ADDR 60000 TO R1
MOV #HODO,CMR ;HODO ALLOWS HIT REGISTER TO BE CLOCKED
;ONLY DURING THE DESTINATION ACCESS
;OF AN INSTRUCTION.
;NO UCB SO AS TO WRITE CACHE STORES

MOV #15,CCR
TST (R0) ;READ HIT
TST (R1) ;READ MISS
TST (R1) ;READ MISS
TST (R1) ;READ MISS
TST (R1) ;READ MISS
TST (R1) ;READ MISS
TST (R1) ;READ MISS
MOV CHR,R2 ;SAVE CHR CONTENTS
NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
NOP ;FOR LOOP ON ERROR
MOV #OFF,CCR ;DISABLE CACHE
CLRB CMR ;DISABLE MAINTENANCE MODE
BIC #177700,R2 ;PREPARE R2 FOR CHECK
CMP #40,R2 ;BIT 05 SHOULD BE 1
BEQ 10$ ;PASS
MOV R2,CHR50 ;PREPARE FOR ERROR REPORT
ERROR ;ERROR
;-----

;CHR BIT 05 DID NOT READ 1 DUE
;TO ONE READ HIT AND 5 READ MISSES
;PRINT CHR<5:0> RECEIVED

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

7127

```

.SBTTL TEST # 225 - EXERCISE CACHE BY READING MEMORY LOCATIONS
*****
*TEST 225 EXERCISE CACHE BY READING MEMORY LOCATIONS
* THIS TEST EXERCISES CACHE BY READING MEMORY LOCATIONS FROM
* 60000 TO 77776 WITH CACHE ON. ALL 4K OF CACHE WILL HAVE BEEN
* EXERCISED. EACH ADDRESS FROM 60000 TO 77776 IS LOADED WITH
* DATA CORRESPONDING TO ITS OWN ADDRESS.
*****
TEST225:
SCPCND ;SCOPE CONDITIONS:GO SET UP FOR LOOP ON
;ERROR/LOOP ON TEST
;TEST START LOCATION
;LOOP ON ERROR START LOCATION
;SCOPE SYNC. NOT IMPLEMENTED FOR THIS TEST
;LOOP ON ERROR END LOCATION

40$:
1$: MOV #60000,R0 ;FIRST ADDRESS
2$: MOV R0,(R0) ;FILL MEMORY WITH ADDRESSES
;
;
;
3$: MOV #60000,R0 ;FIRST ADDRESS
CLR 177746 ;ENABLE CACHE
COM (R0) ;DOUBLE COMPLEMENT DATA AND
COM (R0) ;MAKE SURE IT IS IN THE CACHE
MOV (R0),R5 ;CREATE READ HIT;STORE CACHED DATA IN R5
CMP R5,R0 ;CHECK RESULTS
BEQ 5$ ;PASS
MOV #OFF,CCR ;DISABLE CACHE
MOV R0,FAILAD ;SAVE FAILED ADDRESS
MOV R0,EXDAT6 ;GET EXPECTED DATA
MOV R5,RECDAT ;GET RECEIVED DATA
ERROR ;ERROR
;-----

;WORD .-2
7145 043740 050530 FAILAD ;PRINT FAILED ADDRESS
7146 043742 050504 EXDAT6 ;PRINT EXPECTED DATA
7147 043744 050520 RECDAT ;PRINT RECEIVED DATA
;WORD 0
7148 043746 000000 BR 25$ ;
7149 043750 000404 BR 25$ ;
5$: TST (R0)+ ;NEXT ADDRESS
CMP R0,#77776 ;FINISHED?
BLOS 3$ ;CONTINUE
25$: NOP ;INSTRUCTION 'JMP 1$' PLACED HERE
;FOR LOOP ON ERROR
7154 043766 000240 10$: NOP ;END OF TEST
043770 005237 001472 INC $TESTN ;INCREMENT TEST COUNTER
    
```

```

043640
043640 000004
043642 043652
043644 043652
043646 000000
043650 043762
043652
7128 043652 012700 060000
7129 043656 010010
7130 043660 005720
7131 043662 020027 077776
7132 043666 101773
7133 043670 012700 060000
7134 043674 005037 177746
7135 043700 005110
7136 043702 005110
7137 043704 011005
7138 043706 020500
7139 043710 001420
7140 043712 012737 001015 177746
7141 043720 010037 050530
7142 043724 010037 050504
7143 043730 010537 050520
7144 043734 104413
043736 043734
7145 043740 050530
7146 043742 050504
7147 043744 050520
7148 043746 000000
7149 043750 000404
7150 043752 005720
7151 043754 020027 077776
7152 043760 101747
7153 043762 000240
043764 000240
7154 043766 000240
043770 005237 001472
    
```



```

7199 044134 005037 044240          CLR  $ICNT          :CLEAR PASS ITERATION COUNTER
7200 044140 005237 001474          INC  $PASS          :INCREMENT PASS COUNT
7201 044144 042737 100000 001474    BIC  #100000,$PASS  :DON'T ALLW A NEGATIVE #
7202 044152 104401 044160          TYPE .65$          :TYPE ASCIZ STRING
      044156 000410          BR    64$          :GET OVER THE ASCIZ
      :65$: .ASCIZ <CRLF>/END OF PASS # /
      64$:
7203 044200 013746 001474          MOV  $PASS,-(SP)   :SAVE $PASS FOR TYPEOUT
      044204 104405          TYPDS              :GO TYPE--DECIMAL ASC'I WITH SIGN
7204 044206 104401 002122          TYPE ,$.NULL
7205 044212 013700 000042          MOV  42,R0
7206 044216 001405          BEQ  $DOAGN
7207 044220 000005          RESET
7208 044222 004710          SENDAD: JSR  PC,(R0)
7209 044224 000240          NOP
7210 044226 000240          NOP
7211 044230 000240          NOP
7212 044232 000137 002500          $DOAGN: JMP  BEGIN          :START AGAIN
7213
7214 044236 000012          $TIMES: .WORD 10.
7215 044240 000000          $ICNT: .WORD 0
  
```

7216

```

.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
:*
044242 105737 002111 $TYPE: TSTB $TPFLG ;:IS THERE A TERMINAL?
044246 100002 BPL 1$ ;:BR IF YES
044250 000000 HALT ;:HALT HERE IF NO TERMINAL
044252 000430 BR 3$ ;:LEAVE
044254 010046 1$: MOV RO,-(SP) ;:SAVE RO
044256 017600 000002 MOV @2(SP),RO ;:GET ADDRESS OF ASCIZ STRING
044262 122737 000001 001506 CMPB #APTENV,$ENV ;:RUNNING IN APT MODE
044270 001011 BNE 62$ ;:NO,GO CHECK FOR APT CONSOLE
044272 132737 000100 001507 BITB #APTSPOOL,$ENVM ;:SPOOL MESSAGE TO APT
044300 001405 BEQ 62$ ;:NO,GO CHECK FOR CONSOLE
044302 010037 044312 MOV RO,61$ ;:SETUP MESSAGE ADDRESS FOR APT
044306 004737 001620 JSR PC,$ATY3 ;:SPOOL MESSAGE TO APT
044312 000000 61$: .WORD 0 ;:MESSAGE ADDRESS
044314 132737 000040 001507 62$: BITB #APTCSUP,$ENVM ;:APT CONSOLE SUPPRESSED
044322 001003 BNE 60$ ;:YES,SKIP TYPE OUT
044324 112046 2$: MOVB (RO)+,-(SP) ;:PUSH CHARACTER TO BE TYPED ONTO STACK
044326 001005 BNE 4$ ;:BR IF IT ISN'T THE TERMINATOR
044330 005726 TST (SP)+ ;:IF TERMINATOR POP IT OFF THE STACK
044332 012600 60$: MOV (SP)+,RO ;:RESTORE RO
044334 062716 000002 3$: ADD #2,(SP) ;:ADJUST RETURN PC
044340 000002 RTI ;:RETURN
044342 122716 000011 4$: CMPB #HT,(SP) ;:BRANCH IF <HT>
044346 001430 BEQ 8$
044350 122716 000200 CMPB #CRLF,(SP) ;:BRANCH IF NOT <CRLF>
044354 001006 BNE 5$
044356 005726 TST (SP)+ ;:POP <CR><LF> EQUIV
044360 104401 TYPE ;:TYPE A CR AND LF
044362 002117 $CRLF
044364 105037 044602 CLRB $CHARCNT ;:CLEAR CHARACTER COUNT
044370 000755 BR 2$ ;:GET NEXT CHARACTER
044372 004737 044454 5$: JSR PC,$TYPEC ;:GO TYPE THIS CHARACTER
044376 123726 002110 6$: CMPB $FILLC,(SP)+ ;:IS IT TIME FOR FILLER CHARS.?
044402 001350 BNE 2$ ;:IF NO GO GET NEXT CHAR.
044404 013746 002106 MOV $NULL,-(SP) ;:GET # OF FILLER CHARS. NEEDED
;:AND THE NULL CHAR.
044410 105366 000001 7$: DECB 1(SP) ;:DOES A NULL NEED TO BE TYPED?
044414 002770 BLT 6$ ;:BR IF NO--GO POP THE NULL OFF OF STACK
044416 004737 044454 JSR PC,$TYPEC ;:GO TYPE A NULL
044422 105337 044602 DECB $CHARCNT ;:DO NOT COUNT AS A COUNT
044426 000770 BR 7$ ;:LOOP
:HORIZONTAL TAB PROCESSOR
044430 112716 000040 8$: MOVB #' ,(SP) ;:REPLACE TAB WITH SPACE

```

```
044434 004737 044454          9$: JSR PC,$TYPEC      ;;TYPE A SPACE
044440 132737 000007 044602 BITB #7,$CHARCNT    ;;BRANCH IF NOT AT
044446 C01372          BNE 9$              ;;TAB STOP
044450 005726          TST (SP)+           ;;POP SPACE OFF STACK
044452 000724          BR 2$              ;;GET NEXT CHARACTER
044454          $TYPEC:
044454 105777 135416          TSTB @$TKS          ;;CHAR IN KYBD BUFFER?      ;MJD001
044460 100022          BPL 10$            ;;BR IF NOT                ;MJD001
044462 017746 135412          MOV @$TKB,-(SP)    ;;GET CHAR                  ;MJD001
044466 042716 177600          BIC #177600,(SP)  ;;STRIP EXTRANEIOUS BITS   ;MJD001
044472 122716 000023          CMPB #$XOFF,(SP)  ;;WAS CHAR XOFF           ;MJD001
044476 001012          BNE 102$          ;;BR IF NOT                ;MJD001
044500          101$:
044500 105777 135372          TSTB @$TKS          ;;WAIT FOR CHAR            ;MJD001
044504 100375          BPL 101$          ;;BR IF NOT                ;MJD001
044506 117716 135366          MOVB @$TKB,(SP)    ;;GET CHAR                  ;MJD001
044512 042716 177600          BIC #177600,(SP)  ;;STRIP IT                 ;MJD001
044516 122716 000021          CMPB #$XON,(SP)   ;;WAS IT XON?             ;MJD001
044522 001366          BNE 101$          ;;BR IF NOT                ;MJD001
044524          102$:
044524 005726          TST (SP)+           ;;FIX STACK                ;MJD001
044526          10$:
044526 105777 135350          TSTB @$TPS          ;;WAIT UNTIL PRINTER IS READY ;MJD001
044532 100375          BPL 10$            ;;BR IF NOT                ;MJD001
044534 126627 000002 000021          CMPB 2(SP),#$XON  ;;IS CHARACTER A RANDOM XON? ;RAN001
044542 001420          BEQ $TYPEX        ;;BRANCH IF YES           ;RAN001
044544 116677 000002 135332          MOVB 2(SP),@$TPB  ;;LOAD CHAR TO BE TYPED INTO DATA REG.
044552 122766 000015 000002          CMPB #CR,2(SP)    ;;IS CHARACTER A CARRIAGE RETURN?
044560 001003          BNE 1$            ;;BRANCH IF NO
044562 105037 044602          CLRB $CHARCNT    ;;YES--CLEAR CHARACTER COUNT
044566 000406          BR $TYPEX        ;;EXIT
044570 122766 000012 000002 1$: CMPB #LF,2(SP)      ;;IS CHARACTER A LINE FEED?
044576 001402          BEQ $TYPEX        ;;BRANCH IF YES
044600 105227          INCB (PC)+      ;;COUNT THE CHARACTER
044602 000000          $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
044604 000207          $TYPEX: RTS PC
```

7218

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE
*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS    ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*
*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    ;;CALL FOR TYPEOUT
*$TYPOS: MOV      @ (SP),-(SP)    ;;PICKUP THE MODE
          MOVVB   1(SP), $OFILL   ;;LOAD ZERO FILL SWITCH
          MOVVB   (SP)+, $OMODE+1 ;;NUMBER OF DIGITS TO TYPE
          ADD     #2,(SP)        ;;ADJUST RETURN ADDRESS
          BR      $TYPON
*$TYPOC: MOVVB   #1, $OFILL      ;;SET THE ZERO FILL SWITCH
          MOVVB   #6, $OMODE+1    ;;SET FOR SIX(6) DIGITS
*$TYPON: MOVVB   #5, $OCNT       ;;SET THE ITERATION COUNT
          MOV     R3,-(SP)        ;;SAVE R3
          MOV     R4,-(SP)        ;;SAVE R4
          MOV     R5,-(SP)        ;;SAVE R5
          MOVVB   $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
          NEG     R4
          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
          MOVVB   R4, $OMODE      ;;SAVE IT FOR USE
          MOVVB   $OFILL,R4      ;;GET THE ZERO FILL SWITCH
          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
          CLR     R3             ;;CLEAR THE OUTPUT WORD
          1$:    ROL     R5        ;;ROTATE MSB INTO 'C'
          BR      3$
          2$:    ROL     R5        ;;FORM THIS DIGIT
          ROL     R5
          ROL     R5
          MOV     R5,R3
          3$:    ROL     R3        ;;GET LSB OF THIS DIGIT
          DECB   $OMODE          ;;TYPE THIS DIGIT?
          BPL    7$             ;;BR IF NO
          BIC    #177770,R3     ;;GET RID OF JUNK
          BNE    4$             ;;TEST FOR 0
          TST   R4              ;;SUPPRESS THIS 0?
          BEQ   5$             ;;BR IF YES
          4$:    INC     R4        ;;DON'T SUPPRESS ANYMORE 0'S
          BIS   #'0,R3         ;;MAKE THIS DIGIT ASCII
          5$:    BIS   #' ,R3    ;;MAKE ASCII IF NOT ALREADY
  
```

044606	017646	000000	
044612	116637	000001	045031
044620	112637	045033	
044624	062716	000002	
044630	000406		
044632	112737	000001	045031
044640	112737	000006	045033
044646	112737	000005	045030
044654	010346		
044656	010446		
044660	010546		
044662	113704	045033	
044666	005404		
044670	062704	000006	
044674	110437	045032	
044700	113704	045031	
044704	016605	000012	
044710	005003		
044712	006105		1\$:
044714	000404		
044716	006105		2\$:
044720	006105		
044722	006105		
044724	010503		
044726	006103		3\$:
044730	105337	045032	
044734	100016		
044736	042703	177770	
044742	001002		
044744	005704		
044746	001403		
044750	005204		4\$:
044752	052703	000060	
044756	052703	000040	5\$:

044762	110337	045026		MOVB	R3,8\$::SAVE FOR TYPING
044766	104401	045026		TYPE	.8\$::GO TYPE THIS DIGIT
044772	105337	045030		7\$:	DECB	\$OCNT
044776	003347			BGT	2\$::COUNT BY 1
045000	002402			BLT	6\$::BR IF MORE TO DO
045002	005204			INC	R4	::BR IF DONE
045004	000744			BR	2\$::INSURE LAST DIGIT ISN'T A BLANK
045006	012605			6\$:	MOV	(SP)+,R5
045010	012604			MOV	(SP)+,R4	::RESTORE R5
045012	012603			MOV	(SP)+,R3	::RESTORE R4
045014	016666	000002	000004	MOV	2(SP),4(SP)	::RESTORE R3
045022	012616			MOV	(SP)+,(SP)	::SET THE STACK FOR RETURNING
045024	000002			RTI		::RETURN
045026	C00			8\$:	.BYTE	0
045027	000			.BYTE	0	::STORAGE FOR ASCII DIGIT
045030	000			\$OCNT:	.BYTE	0
045031	000			\$OFILL:	.BYTE	0
045032	000000			\$OMODE:	.WORD	0
						::TERMINATOR FOR TYPE ROUTINE
						::OCTAL DIGIT COUNTER
						::ZERO FILL SWITCH
						::NUMBER OF DIGITS TO TYPE

7220

SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.
*CALL:
*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
*      TYPDS    ;;GO TO THE ROUTINE
$TYPDS:
MOV      R0,-(SP)      ;;PUSH R0 ON STACK
MOV      R1,-(SP)      ;;PUSH R1 ON STACK
MOV      R2,-(SP)      ;;PUSH R2 ON STACK
MOV      R3,-(SP)      ;;PUSH R3 ON STACK
MOV      R5,-(SP)      ;;PUSH R5 ON STACK
MOV      #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
MOV      20(SP),R5      ;;GET THE INPUT NUMBER
BPL     1$             ;;BR IF INPUT IS POS.
NEG     R5             ;;MAKE THE BINARY NUMBER POS.
MOVB    #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.
1$:     CLR      R0      ;;ZERO THE CONSTANTS INDEX
MOV     #5DBLK,R3      ;;SETUP THE OUTPUT POINTER
MOVB    #' ,(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
2$:     CLR      R2      ;;CLEAR THE BCD NUMBER
MOV     $DTBL(R0),R1   ;;GET THE CONSTANT
3$:     SUB     R1,R5    ;;FORM THIS BCD DIGIT
BLT     4$           ;;BR IF DONE
INC     R2           ;;INCREASE THE BCD DIGIT BY 1
BR      3$
4$:     ADD     R1,R5    ;;ADD BACK THE CONSTANT
TST     R2           ;;CHECK IF BCD DIGIT=0
BNE     5$           ;;FALL THROUGH IF 0
TSTB   (SP)         ;;STILL DOING LEADING 0'S?
BMI     7$           ;;BR IF YES
5$:     ASLB   (SP)    ;;MSD?
BCC     6$           ;;BR IF NO
MOVB    1(SP),-1(R3)  ;;YES--SET THE SIGN
6$:     BIS     #'0,R2  ;;MAKE THE BCD DIGIT ASCII
7$:     BIS     #' ,R2  ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB    R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST     (R0)+        ;;JUST INCREMENTING
CMP     R0,#10       ;;CHECK THE TABLE INDEX
BLT     2$           ;;GO DO THE NEXT DIGIT
BGT     8$           ;;GO TO EXIT
MOV     R5,R2        ;;GET THE LSD
BR      6$           ;;GO CHANGE TO ASCII
8$:     TSTB   (SP)+   ;;WAS THE LSD THE FIRST NON-ZERO?
BPL     9$           ;;BR IF NO
9$:     MOVB   -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
CLRB   (R3)         ;;SET THE TERMINATOR
MOV     (SP)+,R5     ;;POP STACK INTO R5
MOV     (SP)+,R3     ;;POP STACK INTO R3
MOV     (SP)+,R2     ;;POP STACK INTO R2
MOV     (SP)+,R1     ;;POP STACK INTO R1
MOV     (SP)+,R0     ;;POP STACK INTO R0
TYPE   ,5DBLK      ;;NOW TYPE THE NUMBER

```

```

045034
045034 010046
045036 010146
045040 010246
045042 010346
045044 010546
045046 012746 020200
045052 016605 000020
045056 100004
045060 005405
045062 112766 000055 000001
045070 005000
045072 012703 045250
045076 112723 000040
045102 005002
045104 016001 045240
045110 160105
045112 002402
045114 005202
045116 000774
045120 060105
045122 005702
045124 001002
045126 105716
045130 100407
045132 106316
045134 103003
045136 116663 000001 177777
045144 052702 000060
045150 052702 000040
045154 110223
045156 005720
045160 020027 000010
045164 002746
045166 003002
045170 010502
045172 000764
045174 105726
045176 100003
045200 116663 177777 177776
045206 105013
045210 012605
045212 012603
045214 012602
045216 012601
045220 012600
045222 104401 045250

```

045226	016666	000002	000004	MOV	7(SP),4(SP)	::ADJUST THE STACK
045234	012616			MOV	(SP)+,(SP)	::RETURN TO USER
045236	000002			RTI		
045240	023420			\$DTBL:	10000.	
045242	001750				1000.	
045244	070144				100.	
045246	000012				10.	
045250				\$DBLK:	.BLKW 4	

7222

```

.SBTTL TTY INPUT ROUTINE
:*****
.ENABL LSB
.DSABL LSB
:*****
*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
*CALL:
:*      RDCHR          ;; INPUT A SINGLE CHARACTER FROM THE TTY
:*      RETURN HERE   ;; CHARACTER IS ON THE STACK
:*                      ;; WITH PARITY BIT STRIPPED OFF
:
045260 011646          $RDCHR: MOV      (SP),-(SP)      ;; PUSH DOWN THE PC
045262 016666 000004 000002 MOV      4(SP),2(SP)      ;; SAVE THE PS
045270 105777 134602 1$:   TSTB     @STKS          ;; WAIT FOR
045274 100375          BPL      1$              ;; A CHARACTER
045276 117766 134576 000004 MOVB    @STKB,4(SP)      ;; READ THE TTY
045304 042766 177600 000004 BIC     #^C<177>,4(SP)  ;; GET RID OF JUNK IF ANY
045312 026627 000004 000023 CMP     4(SP),#23       ;; IS IT A CONTROL-S?
045320 001013          BNE     3$              ;; BRANCH IF NO
045322 105777 134550 2$:   TSTB     @STKS          ;; WAIT FOR A CHARACTER
045326 100375          BPL     2$              ;; LOOP UNTIL ITS THERE
045330 117746 134544          MOVB    @STKB,-(SP)     ;; GET CHARACTER
045334 042716 177600          BIC     #^C177,(SP)    ;; MAKE IT 7-BIT ASCII
045340 022627 000021          CMP     (SP)+,#21      ;; IS IT A CONTROL-Q?
045344 001366          BNE     2$              ;; IF NOT DISCARD IT
045346 000750          BR      1$              ;; YES, RESUME
045350 026627 000004 000021 3$:   CMP     4(SP),#$XON    ;; IS IT A RANDOM XON?
045356 001744          BEQ     1$              ;; BRANCH IF YES
045360 026627 000004 000140 CMP     4(SP),#140     ;; IS IT UPPER CASE?
045366 002407          BLT     4$              ;; BRANCH IF YES
045370 026627 000004 000175 CMP     4(SP),#175    ;; IS IT A SPECIAL CHAR?
045376 003003          BGT     4$              ;; BRANCH IF YES
045400 042766 000040 000004 BIC     #40,4(SP)     ;; MAKE IT UPPER CASE
045406 000002          4$:   RTI              ;; GO BACK TO USER
:*****
*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
*CALL:
:*      RDLIN          ;; INPUT A STRING FROM THE TTY
:*      RETURN HERE   ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
:*                      ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
:
045410 010346          $RDLIN: MOV     R3,-(SP)      ;; SAVE R3
045412 012703 045516 1$:   MOV     #$TTYIN,R3    ;; GET ADDRESS
045416 022703 045526 2$:   CMP     #$TTYIN+8.,R3  ;; BUFFER FULL?
045422 101405          BLOS    4$              ;; BR IF YES
045424 104407          RDCHR   ;; GO READ ONE CHARACTER FROM THE TTY
045426 112613          MOVB    (SP)+,(R3)     ;; GET CHARACTER
045430 122713 000177 10$:  CMPB    #177,(R3)     ;; IS IT A RUBOUT
045434 001003          BNE     3$              ;; SKIP IF NOT
045436 104401 002116 4$:   TYPE    ,SQUES       ;; TYPE A '?'
045442 000763          BR      1$              ;; CLEAR THE BUFFER AND LOOP
045444 111337 045514 3$:   MOVB    (R3),9$      ;; ECHO THE CHARACTER
045450 104401 045514          TYPE    ,9$
045454 122723 000015          CMPB    #15,(R3)+    ;; CHECK FOR RETURN
045460 001356          BNE     2$              ;; LOOP IF NOT RETURN
045462 105063 177777          CLRB   -1(R3)        ;; CLEAR RETURN (THE 15)
045466 104401 002120          TYPE    ,LF         ;; TYPE A LINE FEED
045472 012603          MOV     (SP)+,R3     ;; RESTORE R3

```

```

045474 011646          MOV      (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
045476 016666 000004 000002  MOV      4(SP),2(SP)      ;;      FIRST ASCII CHARACTER ON IT
045504 012766 045516 000004  MOV      #STTYIN,4(SP)
045512 000002          RTI                          ;;RETURN
045514      000          9$: .BYTE 0                ;;STORAGE FOR ASCII CHAR. TO TYPE
045515      000          .BYTE 0                ;;TERMINATOR
045516          .BLKB 8.                ;;RESERVE 8 BYTES FOR TTY INPUT
045526      136      125      015  $CNTLU: .ASCIZ //<15><12>  ;;CONTROL 'U'
045533      136      107      015  $CNTLG: .ASCIZ /G/<15><12>  ;;CONTROL 'G'
045540      015      012      123  $MSWR: .ASCIZ <15><12>/SWR = /
045551      040      116      116  $MNEW: .ASCIZ / NEW = /

```

7224

```

.SBTTL READ AN OCTAL NUMBER FROM THE TTY
:*****
:THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
:CHANGE IT TO BINARY.
:CALL:
:RDOCT                                ;;READ AN OCTAL NUMBER
:RETURN HERE                          ;;LOW ORDER BITS ARE ON TOP OF THE STACK
:                                     ;;HIGH ORDER BITS ARE IN $HIOCT
045562 011646 000004 000002 $RDOCT: MOV (SP),-(SP) ;;PROVIDE SPACE FOR THE
045564 016666                                MOV 4(SP),2(SP) ;;INPUT NUMBER
045572 010046                                MOV R0,-(SP) ;;PUSH R0 ON STACK
045574 010146                                MOV R1,-(SP) ;;PUSH R1 ON STACK
045576 010246                                MOV R2,-(SP) ;;PUSH R2 ON STACK
045600 104410 1$: RDLIN ;;READ AN ASCII LINE
045602 012600 MOV (SP)+,R0 ;;GET ADDRESS OF 1ST CHARACTER
045604 005001 CLR R1 ;;CLEAR DATA WORD
045606 005002 CLR R2
045610 112046 2$: MOVB (R0)+,-(SP) ;;PICKUP THIS CHARACTER
045612 001412 BEQ 3$ ;;IF ZERO GET OUT
045614 006301 ASL R1 ;;*2
045616 006102 ROL R2 ;;*4
045620 006301 ASL R1 ;;*8
045622 006102 ROL R2
045624 006301 ASL R1
045626 006102 ROL R2
045630 042716 177770 BIC #^C7,(SP) ;;STRIP THE ASCII JUNK
045634 062601 ADD (SP)+,R1 ;;ADD IN THIS DIGIT
045636 000764 BR 2$ ;;LOOP
045640 005726 3$: TST (SP)+ ;;CLEAN TERMINATOR FROM STACK
045642 010166 000012 MOV R1,12(SP) ;;SAVE THE RESULT
045646 010237 045662 MOV R2,$HIOCT
045652 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
045654 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
045656 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
045660 000002 RTI ;;RETURN
045662 000000 $HIOCT: .WORD 0 ;;HIGH ORDER BITS GO HERE
  
```

7226

```

.SBTTL READ A DECIMAL NUMBER FROM THE TTY
*****
*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
*ARE READ A '?' FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
*POSITIVE 32767 TO NEGATIVE 32768.
*CALL:
**      RDDEC          ::READ A DECIMAL NUMBER
**      RETURN HERE   ::NUMBER IS ON TOP OF THE STACK
.
045664 011646          $RDDEC: MOV      (SP),-(SP)      ::PROVIDE SPACE FOR
045666 016666 000004 000002 MOV      4(SP),2(SP)      ::THE INPUT NUMBER
045674 010046          MOV      R0,-(SP)      ::PUSH R0 ON STACK
045676 010146          MOV      R1,-(SP)      ::PUSH R1 ON STACK
045700 010246          MOV      R2,-(SP)      ::PUSH R2 ON STACK
045702 104410          1$:  RDLIN          ::READ AN ASCII LINE
045704 012600          MOV      (SP)+,R0      ::ADDRESS OF 1ST CHAR.
045706 010037 046032  MOV      R0,6$      ::SAVE INCASE OF BAD INPUT
045712 005046          CLR      -(SP)      ::CLEAR DATA WORD
045714 005002          CLR      R2          ::SIGN SET POSITIVE
045716 122710 000055  CMPB     #'-',(R0)      ::SEE IF A MINUS SIGN WAS TYPED
045722 001001          BNE     2$          ::BR IF NO MINUS SIGN
045724 112002          MOVB    (R0)+,R2      ::SAVE FOR LATER USE
045726 112001          2$:  MOVB    (R0)+,R1      ::PICKUP THIS CHARACTER
045730 001424          BEQ     3$          ::GET OUT IF ZERO
045732 122701 000060  CMPB     #'0',R1      ::MAKE SURE THIS CHARACTER
045736 003032          BGT     5$          ::IS A DIGIT BETWEEN 0 & 9
045740 122701 000071  CMPB     #'9',R1
045744 002427          BLT     5$
045746 032716 170000  BIT      #'C7777',(SP)  ::DON'T LET NUMBER GET TO BIG
045752 001024          BNE     5$          ::BR IF NUMBER WOULD OVERFLOW
045754 006316          ASL     (SP)        ::*2
045756 011646          MOV     (SP),-(SP)   ::SAVE FOR LATER
045760 006316          ASL     (SP)        ::*4
045762 006316          ASL     (SP)        ::*8
045764 062616          ADD     (SP)+,(SP)   ::*10
045766 102416          BVS     5$          ::OVERFLOW ISN'T ALLOWED
045770 162701 000060  SUB     #'0',R1      ::STRIP AWAY THE ASCII JUNK
045774 060116          ADD     R1,(SP)     ::ADD IN THIS DIGIT
045776 102412          BVS     5$          ::OVERFLOW ISN'T ALLOWED
046000 000752          BR      2$          ::LOOP
046002 005702          3$:  TST     R2          ::CHECK IF NUMBER IS NEG
046004 001401          BEQ     4$          ::BR IF NO
046006 005416          NEG     (SP)        ::YES--NEGATE THE NUMBER
046010 012666 000012  4$:  MOV     (SP)+,12(SP)  ::SAVE THE RESULT
046014 012602          MOV     (SP)+,R2    ::POP STACK INTO R2
046016 012601          MOV     (SP)+,R1    ::POP STACK INTO R1
046020 012600          MOV     (SP)+,R0    ::POP STACK INTO R0
046022 000002          RTI          ::RETURN
046024 005726          5$:  TST     (SP)+      ::CLEAN PARTIAL NUMBER FROM STACK
046026 105010          CLRB    (R0)        ::SET A TERMINATOR
046030 104401          TYPE          ::TYPE THE INPUT UP TO BAD CHAR.
046032 000000          6$:  .WORD    0          ::POINTER GOES HERE
046034 104401 002116  TYPE     ,SQUES      ::"'?' 'CR' & 'LF'
046040 000720          BR      1$          ::TRY AGAIN
  
```

7228

```

.SBTTL BINARY TO ASCII AND TYPE ROUTINE
:*****
:THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
: BINARY-ASCII NUMBER AND TYPE IT.
:CALL:
:  MOV      NUMBER,-(SP)      ;;NUMBER TO BE TYPED
:  TYPBN
:STYPBN: MOV      R1,-(SP)      ;;SAVE R1 ON THE STACK
:        MOV      6(SP),R1     ;;GET THE INPUT NUMBER
:        SEC
:        MOVB     #'0,$BIN     ;;SET 'C' SO CAN KEEP TRACK OF THE NUMBER OF BITS
:        ROL      R1           ;;GET THIS BIT
:        BEQ      2$           ;;DONE?
:        ADCB     $BIN         ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
:        TYPE     ,$BIN        ;;GO TYPE THIS BIT
:        CLC
:        BR       1$           ;;CLEAR 'C' SO CAN KEEP TRACK OF BITS
:        BR       1$           ;;GO DO THE NEXT BIT
:        MOV      (SP)+,R1     ;;POP THE STACK INTO R1
:        MOV      2(SP),4(SP)  ;;ADJUST THE STACK
:        MOV      (SP)+,(SP)
:        RTI
:        $BIN: .BYTE 0,0      ;;RETURN TO USER
:                                     ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
046042 010146
046044 016601 000006
046050 000261
046052 112737 000060 046114 1$:
046060 006101
046062 001406
046064 105537 046114
046070 104401 046114
046074 000241
046076 000765
046100 012601
046102 016666 000002 000004 2$:
046110 012616
046112 000002
046114 000 000 $BIN: .BYTE 0,0

```

TRAP

.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.

046116 010046
046120 016600 000002
046124 005740
046126 111000
046130 006300
046132 016000 046152
046136 000200

\$TRAP: MOV R0,-(SP) ;;SAVE R0
MOV 2(SP),R0 ;;GET TRAP ADDRESS
TST -(R0) ;;BACKUP BY 2
MOVB (R0),R0 ;;GET RIGHT BYTE OF TRAP
ASL R0 ;;POSITION FOR INDEXING
MOV \$TRPAD(R0),R0 ;;INDEX TO TABLE
RTS R0 ;;GO TO ROUTINE

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

046140 011646
046142 016666 000004 000002
046150 000002

\$TRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
RTI ;;RESTORE THE PSW

.SBTTL TRAP TABLE

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

046152 046140
046154 044242
046156 044632
046160 044606
046162 044646
046164 045034
046166 046042
046170 045260
046172 045410
046174 045562
046176 045664
7231 046200 046224

\$TRPAD: .WORD \$TRAP2
\$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
\$TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
\$TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
\$TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
\$TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
\$TYPBN ;;CALL=TYPBN TRAP+6(104406) TYPE BINARY (ASCII) NUMBER
\$RDCHR ;;CALL=RDCHR TRAP+7(104407) TTY TYPEIN CHARACTER ROUTINE
\$RDLIN ;;CALL=RDLIN TRAP+10(104410) TTY TYPEIN STRING ROUTINE
\$RDOCT ;;CALL=RDOCT TRAP+11(104411) READ AN OCTAL NUMBER FROM TTY
\$RDDEC ;;CALL=RDDEC TRAP+12(104412) READ A DECIMAL NUMBER FROM TTY
\$ERROR ;;CALL=ERROR TRAP+13(104413)


```
7233                                     .SBTTL  ERROR HANDLER ROUTINE
7234 :.....
7235 046202 000000  ERRPC:  .WORD  0
7236 046204 000000  SAVR0:  .WORD
7237 046206 000000  SAVR1:  .WORD
7238 046210 000000  SAVR2:  .WORD
7239 046212 000000  SAVR3:  .WORD
7240 046214 000000  SAVR4:  .WORD
7241 046216 000000  SAVR5:  .WORD
7242 046220 000000  CPSAVE: .WORD  0      ;LOCATION TO SAVE CPU ERROR REG CONTENTS
7243 046222 000000  IBSAVE: .WORD  0      ;FLAG TO INDICATE 2ND ERROR CALL DUE
```

```

7244
7245 046224 105037 046222      $ERROR: CLRB      IBSAVE      ;CLEAR THE ITFM BYTE SAVE LOCATION      ;DPM001
7246 046230 010037 046204      $EROVR: MOV       R0,SAVR0      ;SAVE R0 THRU R5
7247 046234 010137 046206      MOV       R1,SAVR1
7248 046240 010237 046210      MOV       R2,SAVR2
7249 046244 010337 046212      MOV       R3,SAVR3
7250 046250 010437 046214      MOV       R4,SAVR4
7251 046254 010537 046216      MOV       R5,SAVR5
7252 046260 113777 002060 133606      MOVVB    $TSTNM,@SWR      ;MOVE TEST NUMBER TO DISPLAY FOR ALL TO SEE
7253 046266 032777 020000 133600      BIT      #BIT13,@SWR      ;INHIBIT ERROR TYPEOUTS?
7254 046274 001160      BNE      99$              ;YES
7255 046276 011601      MOV      (SP),R1          ;R1 CONTAINS ADDRESS FOLLOWING ERRPC ADDRESS
7256 046300 012137 046202      MOV      (R1)+,ERRPC      ;LOAD ERRPC ADDRESS AND POINT TO NEXT ARGUMENT
7257 046304 022711 046220      CMP      #CPSAVE,(R1)     ;SEE IF ORIG CALL WAS A PWR MNR BIT ERR ;DPM001
7258 046310 001431      BEQ      1001$           ;BRANCH IF SO - NEXT TEST IS SUPERFLUOUS;DPM001
7259 046312 105737 046222      TSTB    IBSAVE           ;SEE IF THIS IS THE 2ND ERROR CALL      ;DPM001
7260 046316 001024      BNE      1000$           ;BRANCH IF SO                          ;DPM001
7261 046320 013737 177766 046220      MOV      177766,CPSAVE    ;MOVE CPU ERR REG TO CPSAVE FOR TEST    ;DPM001
7262 046326 032737 000001 046220      BIT      #BIT00,CPSAVE    ;SEE IF POWER MONITOR BIT IS SET        ;DPM001
7263 046334 001417      BEQ      1001$           ;BRANCH IF OK                          ;DPM001
7264 046336 042737 000001 177766      BIC      #BIT00,177766    ;CLEAR THE BIT FOUND SET                ;DPM001
7265 046344 012737 000177 001470      MOV      #177,$FATAL     ;LET APT KNOW THIS IS PWR MNR BIT ERR   ;DPM001
7266 046352 105237 046222      INCB    IBSAVE           ;MAKE IBSAVE NON-ZERO FOR DUAL CALL     ;DPM001
7267 046356 012701 046364      MOV      #500$,R1        ;MOVE START LOCATIONS OF POINTERS TO R1 ;DPM001
7268 046362 000404      BR      1001$           ;BRANCH OVER IBSAVE CLEARING            ;DPM001
7269 046364 046220 000000      .WORD   CPSAVE,0         ;1 DATA WORD TO PRINT                  ;DPM001
7270 046370 105037 046222      1000$: CLRB    IBSAVE     ;CLEAR IBSAVE SO AFTER 2ND ERROR, EX11 ;DPM001
7271 046374 117737 177602 001470 1001$: MOVVB    @ERRPC,$FATAL   ;LOAD $FATAL FOR APT                    ;DPM001
7272 046402 104401 046410      TYPE    .65$            ;:TYPE ASCIZ STRING
046406 000411      BR      64$              ;:GET OVER THE ASCIZ
::65$: .ASCIZ <CRLF><CRLF>/TESTNO ERRPC/
64$:
7273 046432 005711      3$:      TST      (R1)          ;END OF ARGUMENTS?
7274 046434 001412      BEQ      25$            ;YES,GO PRINT DATA
7275 046436 012702 050420      MOV      #PRTABL,R2     ;ADDRESS OF START OF PRINT TABLE LIST
7276 046442 012703 047056      MOV      #PRTITL,R3     ;ADDRESS OF START OF TITLES
7277 046446 005723      2$:      TST      (R3)+          ;INDEX THRU TITLES
7278 046450 021122      CMP      (R1),(R2)+     ;SEARCH PRINT TABLE LIST FOR TITLE
7279 046452 001375      BNE      2$             ;NO; CHECK NEXT LOCATION IN LIST
7280 046454 004753      JCR     PC,@-(R3)       ;FOUND IT; GO PRINT TITLE
7281 046456 005721      TST      (R1)+          ;R1 POINTS TO NXT ARGUMENT IN TEST CODE
7282 046460 000764      BR      3$
7283 046462      25$:     TYPE    .67$            ;:TYPE ASCIZ STRING
046462 000401 046470      BR      66$            ;:GET OVER THE ASCIZ
046466      ::67$: .ASCIZ <CRLF>
66$:
7284 046472      MOV      $TESTN,-(SP)    ;:SAVE $TESTN FOR TYPEOUT
046476 104403      TYPOS   ;:GO TYPE--OCTAL ASCII
046500 006      .BYTE   6              ;:TYPE 6 DIGIT(S)
046501 000      .BYTE   0              ;:SUPPRESS LEADING ZEROS
7285 046502 104401 046510      TYPE    .69$            ;:TYPE ASCIZ STRING
046506 000402      BR      68$            ;:GET OVER THE ASCIZ
::69$: .ASCIZ / /
68$:
7286 046514      MOV      ERRPC,-(SP)    ;:SAVE ERRPC FOR TYPEOUT
046520 104402      TYPOC   ;:GO TYPE--OCTAL ASCII(ALL DIGITS)

```

```

7287 046522 104401 002122          TYPE      , $ENULL
7288 046526 022701 046366          CMP       #500$,R1      ;SEE IF THIS IS SPECIAL PWR MNTR BIT ERR;DPM001
7289 046532 001003                    BNE      29$          ;BRANCH IF NOT ;DPM001
7290 046534 012701 046364          MOV       #500$,R1    ;RESET POINTER ;DPM001
7291 046540 000734                    BR       3$          ;BRANCH OVER NON-PWR MNTR BIT ERR SETUP ;DPM001
7292 046542 011601          29$:    MOV      (SP),R1     ;R1 CONTAINS ADDRESS FOLLOWING ERRORPC ADDRESS
7293 046544 005721                    TST      (R1)+       ;POINT TO NEXT ARGUMENT
7294 046546 005711          13$:    TST      (R1)       ;END OF ARGUMENTS?
7295 046550 001432                    BEQ      99$         ;YES
7296 046552 012702 050420          MOV      #PRTABL,R2  ;ADDRESS OF START OF PRINT TABLE LIST
7297 046556 012703 050064          MOV      #PRDATA,R3 ;ADDRESS OF START OF PRINT DATA
7298 046562 005723          12$:    TST      (R3)+       ;INDEX THRU DATA PRINTS
7299 046564 021122                    CMP      (R1),(R2)+  ;SEARCH PRINT TABLE LIST FOR TITLE
7300 046566 001375                    BNE      12$         ;NO; CHECK NEXT LOCATION IN LIST
7301 046570 104401 046576          TYPE     ,71$       ;:TYPE ASCIZ STRING
7301 046574 000404                    BR       70$         ;:GET OVER THE ASCIZ
;:71$: .ASCIZ / /
70$:
046606
7302 046606 004753                    JSR      PC,@-(R3)   ;
7303 046610 104401 002122          TYPE     , $ENULL
7304 046614 104401 046622          TYPE     ,73$       ;:TYPE ASCIZ STRING
7304 046620 000404                    BR       72$         ;:GET OVER THE ASCIZ
;:73$: .ASCIZ / /
72$:
7305 046632 005721                    TST      (R1)+       ;R1 POINTS TO NEXT ARGUMENT
7306 046634 000744                    BR       13$
7307
7308
;INHIBIT ERROR TYPEOUT CODE
7309 046636 011601          99$:    MOV      (SP),R1     ;R1 CONTAINS ADDRESS FOLLOWING ERRPC ADDRESS
7310 046640 005711          111$:   TST      (R1)       ;IS THIS THE END OF ARGUMENT LIST?
7311 046642 001402                    BEQ      112$        ;YES
7312 046644 005721                    TST      (R1)+       ;POINT TO NEXT ARGUMENT
7313 046646 000774                    BR       111$
7314 046650 005721          112$:   TST      (R1)+       ;R1 NOW CONTAINS RETURN ADDRESS
7315 046652 022701 046370          CMP      #1000$,R1  ;SEE IF PWR MNTR BIT ERROR ;DPM001
7316 046656 001401                    BEQ      100$        ;BRANCH OVER NEXT INST IF SO ;DPM001
7317 046660 010116                    MOV      R1,(SP)    ;SETUP RETURN ADDRESS IN STACK
7318
7319
;HOE,LOE OPTION DETERMINATION
7320
7321 046662 122737 000001 001506 100$:  CMPB     #APTENV,$ENV ;IS THIS APT?
7322 046670 001410                    BEQ      52$         ;YES HALT ON ERROR
7323 046672 023737 000042 000046          CMP      42,46      ;IS THIS ACT. ; QV OR AUTO ACCEPT
7324 046700 001404                    BEQ      52$         ;YES HALT ON ERROR
7325 046702 032777 100000 133164          BIT      #BIT15,@SWR ;IS HALT ON ERROR IMPLEMENTED?
7326 046710 001404                    BEQ      51$         ;NO
7327 046712 012737 000001 001466 52$:    MOV      #1,$MSGTY   ;SET $MSGTY FOR POSSIBLE APT USE
7328 046720 000000                    HALT
7329 046722 032777 001000 133144 51$:    BIT      #BIT09,@SWR ;IS LOOP ON ERROR IMPLEMENTED?
7330 046730 001007                    BNE      54$         ;YES
7331
7332
;NO; INITIALIZE LOCATIONS WHERE
7333
; 'JMP 1$' IS PLACED FOR LOOP ON ERROR
; WITH NOP'S
7334 046732 012777 000240 133174          MOV      #240,@ADJRMP
7335 046740 012777 000240 133170          MOV      #240,@ADR1$
7336 046746 000415                    BR       55$
7337 046750 012777 000137 133156 54$:    MOV      #137,@ADJRMP ;CONTINUE WITH PRESENT TEST
;WRITE 'JMP' INSTRUCTION TO PROPER ADDRESS

```

```

7338 046756 013777 002130 133152      MOV      STRTLP,@ADR1$  ;WRITE '1$' LOCATION TO PROPER ADDRESS
7339 046764 013737 002132 177752      MOV      ADRSYNC,CHR   ;LOAD ADDRESS LOCATION FOR SCOPE SYNC
7340 046772 105037 177751      CLR      CLR          ;
7341 046776 013716 002130      MOV      STRTLP,(SP)   ;SETUP LOCATION FOR LOOP ON ERROR IN STACK
7342 047002 013700 046204      55$:    MOV      SAVR0,R0     ;RESTORE REGISTERS
7343 047006 013701 046206      MOV      SAVR1,R1
7344 047012 013702 046210      MOV      SAVR2,R2
7345 047016 013703 046212      MOV      SAVR3,R3
7346 047022 013704 046214      MOV      SAVR4,R4
7347 047026 013705 046216      MOV      SAVR5,R5
7348 047032 032777 001000 133034      BIT      #BIT09,@SWR   ;CHECK TO SEE IF LOOP ON ERROR ENABLED ;DPM001
7349 047040 001005          BNE      6$           ;BRANCH IF SO - DON'T CHECK FOR 2ND ERR ;DPM001
7350 047042 005737 046222      TST      IBSAVE       ;SEE IF THIS IS 1ST OF 2 ERRORS TO CALL ;DPM001
7351 047046 001402          BEQ      6$           ;BRANCH AROUND RETURN JUMP IF NOT ;DPM001
7352 047050 000137 046230      JMP      $EROVR       ;JUMP BACK TO CALL 2ND ERROR ;DPM001
7353 047054 000002          6$:      RTI           ;RETURN
7354
7355 047056 047122 047156 047212 PRTITL: .WORD 1$,2$,3$,4$,5$,6$,7$,8$,9$,10$,11$,12$,13$,14$,15$,16$,17$,18$
7356 047122 104401 047130      1$:      TYPE      ,65$      ;:TYPE ASCIZ STRING
          BR      64$      ;:GET OVER THE ASCIZ
          ;:65$: .ASCIZ / CA210(21:0) /
          64$:
7357 047154 000207          RTS PC
7358 047156 104401 047164      2$:      TYPE      ,67$      ;:TYPE ASCIZ STRING
          BR      66$      ;:GET OVER THE ASCIZ
          ;:67$: .ASCIZ / AMR210(21:0) /
          66$:
7359 047210 000207          RTS PC
7360 047212 104401 047220      3$:      TYPE      ,69$      ;:TYPE ASCIZ STRING
          BR      68$      ;:GET OVER THE ASCIZ
          ;:69$: .ASCIZ / CHR157(15:07) /
          68$:
7361 047244 000207          RTS PC
7362 047246 104401 047254      4$:      TYPE      ,71$      ;:TYPE ASCIZ STRING
          BR      70$      ;:GET OVER THE ASCIZ
          ;:71$: .ASCIZ / CA2113(21:13) /
          70$:
7363 047300 000207          RTS PC
7364 047302 104401 047310      5$:      TYPE      ,73$      ;:TYPE ASCIZ STRING
          BR      72$      ;:GET OVER THE ASCIZ
          ;:73$: .ASCIZ / CHR80(8:0) /
          72$:
7365 047334 000207          RTS PC
7366 047336 104401 047344      6$:      TYPE      ,75$      ;:TYPE ASCIZ STRING
          BR      74$      ;:GET OVER THE ASCIZ
          ;:75$: .ASCIZ / CDR150(15:0) /
          74$:
7367 047370 000207          RTS PC
7368 047372 104401 047400      7$:      TYPE      ,77$      ;:TYPE ASCIZ STRING
          BR      76$      ;:GET OVER THE ASCIZ
          ;:77$: .ASCIZ /

```

```
047424  
7369 047424 000207  
7370 047426 104401 047434  
047426 000412  
047432  
  
047460  
7371 047460 000207  
7372 047462 104401 047470  
047462 000412  
047466  
  
047514  
7373 047514 000207  
7374 047516 104401 047524  
047516 000412  
047522  
  
047550  
7375 047550 000207  
7376 047552 104401 047560  
047552 000412  
047556  
  
047604  
7377 047604 000207  
7378 047606 104401 047614  
047606 000412  
047612  
  
047640  
7379 047640 000207  
7380 047642 104401 047650  
047642 000406  
047646  
  
047664  
7381 047664 000207  
7382 047666 104401 047674  
047666 000411  
047672  
  
047716  
7383 047716 000207  
7384 047720 104401 047726  
047720 000407  
047724  
  
047744  
7385 047744 000207  
7386 047746 104401 047754  
047746 000411  
047752  
  
047776  
7387 047776 000207
```

```
77$: .ASCIZ / EXDAT6 /  
76$:  
8$: RTS PC  
TYPE 79$ ::TYPE ASCIZ STRING  
BR 78$ ::GET OVER THE ASCIZ  
79$: .ASCIZ / CA121(12:1) /  
78$:  
9$: RTS PC  
TYPE 81$ ::TYPE ASCIZ STRING  
BR 80$ ::GET OVER THE ASCIZ  
81$: .ASCIZ / EXDAT1 /  
80$:  
10$: RTS PC  
TYPE 83$ ::TYPE ASCIZ STRING  
BR 82$ ::GET OVER THE ASCIZ  
83$: .ASCIZ / CM1513(15:13) /  
82$:  
11$: RTS PC  
TYPE 85$ ::TYPE ASCIZ STRING  
BR 84$ ::GET OVER THE ASCIZ  
85$: .ASCIZ / CNT121(12:1) /  
84$:  
12$: RTS PC  
TYPE 87$ ::TYPE ASCIZ STRING  
BR 86$ ::GET OVER THE ASCIZ  
87$: .ASCIZ / FLTPAT /  
86$:  
13$: RTS PC  
TYPE 89$ ::TYPE ASCIZ STRING  
BR 88$ ::GET OVER THE ASCIZ  
89$: .ASCIZ / RECDAT /  
88$:  
14$: RTS PC  
TYPE 91$ ::TYPE ASCIZ STRING  
BR 90$ ::GET OVER THE ASCIZ  
91$: .ASCIZ / EXDAT3 /  
90$:  
15$: RTS PC  
TYPE 93$ ::TYPE ASCIZ STRING  
BR 92$ ::GET OVER THE ASCIZ  
93$: .ASCIZ / CHR50 /  
92$:  
16$: RTS PC  
TYPE 95$ ::TYPE ASCIZ STRING  
BR 94$ ::GET OVER THE ASCIZ  
95$: .ASCIZ / CMR119 /  
94$:  
RTS PC
```

```
7388 050000 104401 050006 17$: TYPE ,97$ ;;TYPE ASCIZ STRING
      050000 000411 050006 BR ,96$ ;;GET OVER THE ASCIZ
      050004 000411 .ASCIZ / FAILAD /
      050030 96$: RTS PC
7389 050030 000207 18$: TYPE ,99$ ;;TYPE ASCIZ STRING
7390 050032 104401 050040 BR ,98$ ;;GET OVER THE ASCIZ
      050036 000411 .ASCIZ / CPUERR /
      050062 98$: RTS PC
7391 050062 000207 PRDATA: .WORD 1$,2$,3$,4$,5$,6$,7$,8$,9$,10$,11$,12$,13$,14$,15$,16$,17$,18$
7392 050064 050130 050152 050174 1$:
7393 050130 013746 050464 MOV CA210,-(SP) ;;SAVE CA210 FOR TYPEOUT
      050134 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050136 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
      050137 001 .BYTE 1 ;;TYPE LEADING ZEROS
7394 050140 013746 050466 MOV CA210+2,-(SP) ;;SAVE CA210+2 FOR TYPEOUT
      050144 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050146 005 .BYTE 5 ;;TYPE 5 DIGIT(S)
      050147 001 .BYTE 1 ;;TYPE LEADING ZEROS
7395 050150 000207 2$:
7396 050152 013746 050470 MOV AMR210,-(SP) ;;SAVE AMR210 FOR TYPEOUT
      050156 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050160 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
      050161 001 .BYTE 1 ;;TYPE LEADING ZEROS
7397 050162 013746 05047 MOV AMR210+2,-(SP) ;;SAVE AMR210+2 FOR TYPEOUT
      050166 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050170 005 .BYTE 5 ;;TYPE 5 DIGIT(S)
      050171 001 .BYTE 1 ;;TYPE LEADING ZEROS
7398 050172 000207 3$:
7399 050174 013746 050474 MOV CHR157,-(SP) ;;SAVE CHR157 FOR TYPEOUT
      050200 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050202 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
      050203 001 .BYTE 1 ;;TYPE LEADING ZEROS
7400 050204 000207 4$:
7401 050206 013746 050476 MOV CA2113,-(SP) ;;SAVE CA2113 FOR TYPEOUT
      050212 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050214 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
      050215 001 .BYTE 1 ;;TYPE LEADING ZEROS
7402 050216 000207 5$:
7403 050220 013746 050500 MOV CHR80,-(SP) ;;SAVE CHR80 FOR TYPEOUT
      050224 104403 TYPOS ;;GO TYPE--OCTAL ASCII
      050226 003 .BYTE 3 ;;TYPE 3 DIGIT(S)
      050227 001 .BYTE 1 ;;TYPE LEADING ZEROS
7404 050230 000207 6$:
7405 050232 013746 050502 MOV CDR150,-(SP) ;;SAVE CDR150 FOR TYPEOUT
      050236 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
      050240 000207 RTS PC
7406 050242 013746 050504 7$:
7407 050242 013746 050504 MOV EXDAT6,-(SP) ;;SAVE EXDAT6 FOR TYPEOUT
```

```

050246 104402          TYPOC          ::GO TYPE--OCTAL ASCII(ALL DIGITS)
7408 050250 000207    RTS PC
7409 050252          8$:
050252 013746 050506  MOV      CA121,-(SP)  ::SAVE CA121 FOR TYPEOUT
050256 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050260      004        .BYTE      4        ::TYPE 4 DIGIT(S)
050261      001        .BYTE      1        ::TYPE LEADING ZEROS
7410 050262 000207    RTS PC
7411 050264          9$:
050264 013746 050510  MOV      EXDAT1,-(SP) ::SAVE EXDAT1 FOR TYPEOUT
050270 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050272      001        .BYTE      1        ::TYPE 1 DIGIT(S)
050273      001        .BYTE      1        ::TYPE LEADING ZEROS
7412 050274 000207    RTS PC
7413 050276          10$:
050276 013746 050512  MOV      CM1513,-(SP) ::SAVE CM1513 FOR TYPEOUT
050302 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050304      001        .BYTE      1        ::TYPE 1 DIGIT(S)
050305      001        .BYTE      1        ::TYPE LEADING ZEROS
7414 050306 000207    RTS PC
7415 050310          11$:
050310 013746 050514  MOV      CNT121,-(SP) ::SAVE CNT121 FOR TYPEOUT
050314 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050316      004        .BYTE      4        ::TYPE 4 DIGIT(S)
050317      001        .BYTE      1        ::TYPE LEADING ZEROS
7416 050320 000207    RTS PC
7417 050322          12$:
050322 013746 050516  MOV      FLTPAT,-(SP) ::SAVE FLTPAT FOR TYPEOUT
050326 104402          TYPOC          ::GO TYPE--OCTAL ASCII(ALL DIGITS)
7418 050330 000207    RTS PC
7419 050332          13$:
050332 013746 050520  MOV      RECDAT,-(SP) ::SAVE RECDAT FOR TYPEOUT
050336 104402          TYPOC          ::GO TYPE--OCTAL ASCII(ALL DIGITS)
7420 050340 000207    RTS PC
7421 050342          14$:
050342 013746 050522  MOV      EXDAT3,-(SP) ::SAVE EXDAT3 FOR TYPEOUT
050346 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050350      003        .BYTE      3        ::TYPE 3 DIGIT(S)
050351      001        .BYTE      1        ::TYPE LEADING ZEROS
7422 050352 000207    RTS PC
7423 050354          15$:
050354 013746 050524  MOV      CHR50,-(SP)  ::SAVE CHR50 FOR TYPEOUT
050360 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050362      002        .BYTE      2        ::TYPE 2 DIGIT(S)
050363      001        .BYTE      1        ::TYPE LEADING ZEROS
7424 050364 000207    RTS PC
7425 050366          16$:
050366 013746 050526  MOV      CMR119,-(SP) ::SAVE CMR119 FOR TYPEOUT
050372 104403          TYPOS          ::GO TYPE--OCTAL ASCII
050374      001        .BYTE      1        ::TYPE 1 DIGIT(S)
050375      001        .BYTE      1        ::TYPE LEADING ZEROS
7426 050376 000207    RTS PC
7427 050400          17$:
050400 013746 050530  MOV      FAILAD,-(SP) ::SAVE FAILAD FOR TYPEOUT
050404 104402          TYPOC          ::GO TYPE--OCTAL ASCII(ALL DIGITS)
7428 050406 000207    RTS PC
7429 050410          18$:

```

050410 013746 046220
050414 104402
7430 050416 000207

MOV (PSAVE, -(SP) ;;SAVE (PSAVE FOR TYPEOUT
TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
RTS PC


```

7432 050420 050464 050470 050474 PRTABL: .WORD CA210,AMR210,CHR157,CA2113,CHR80,CDR150,EXDAT6,CA121,EXDAT1,CM1513
7433 050444 050514 050516 050520 .WORD CNT121,FLTPAT,RECDAT,EXDAT3,CHR50,CMR119,FAILAD,CPSAVE
7434
7435 050464 000000 000000 CA210: .WORD 0,0
7436 050470 000000 000000 AMR210: .WORD 0,0
7437 050474 000000 CHR157: .WORD 0
7438 050476 000000 CA2113: .WORD 0
7439 050500 000000 CHR80: .WORD 0
7440 050502 000000 CDR150: .WORD 0
7441 050504 000000 EXDAT6: .WORD 0
7442 050506 000000 CA121: .WORD 0
7443 050510 000000 EXDAT1: .WORD 0
7444 050512 000000 CM1513: .WORD 0
7445 050514 000000 CNT121: .WORD 0
7446 050516 000000 FLTPAT: .WORD 0
7447 050520 000000 RECDAT: .WORD 0
7448 050522 000000 EXDAT3: .WORD 0
7449 050524 000000 CHR50: .WORD 0
7450 050526 000000 CMR119: .WORD 0
7451 050530 000000 FAILAD: .WORD 0
7452
7453 060000 .:=60000
7454 060000 LOW1: .WORD 0
7455 070000 .:=70000
7456 070000 HIGH1: .WORD 0
7457 000001 .END

```

ABASE = 000000	BEBA = 170004	FAIL1 002070	SR3 = 172516	TST147 025620
ACDW1 = 000000	BECC = 170002	FAIL2 002072	STAR* 001000	TST15 003746
ACDW2 = 000000	BECR1 = 170006	FC = 000400	STRTP 002130	TST150 026140
ACPUOP = 000000	BECR2 = 170016	FLTPAT 050516	STRST 002126	TST151 026534
ADDW0 = 000000	BEDA = 170000	FMI = 000010	SWR 002074	TST152 027130
ADDW1 = 000000	BEGIN 002500	FMLO = 000004	SWREG 000176	TST153 027272
ADDW10 = 000000	BIT00 = 000001	HIGH1 070000	TDAR = 000001	TST154 027464
ADDW11 = 000000	BIT01 = 000002	HIT = 000400	TPB = 001000	TST155 027650
ADDW12 = 000000	BIT02 = 000004	HODO = 000002	TPE = 000040	TST156 030030
ADDW13 = 000000	BIT03 = 000010	HPB = 004000	TSTCNT 002140	TST157 030156
ADDW14 = 000000	BIT04 = 000020	HT = 000011	TSTID = 000001	TST16 004016
ADDW15 = 000000	BIT05 = 000040	IBSAVE 046222	TSTIMS 002142	TST160 030332
ADDW2 = 000000	BIT06 = 000100	KPAR0 = 172340	TST1 002576	TST161 030476
ADDW3 = 000000	BIT07 = 000200	KPAR1 = 172342	TST10 003364	TST162 030726
ADDW4 = 000000	BIT08 = 000400	KPAR2 = 172344	TST100 012730	TST163 031156
ADDW5 = 000000	BIT09 = 001000	KPAR3 = 172346	TST101 013154	TST164 031412
ADDW6 = 000000	BIT10 = 002000	KPAR4 = 172350	TST102 013336	TST165 031646
ADDW7 = 000000	BIT11 = 004000	KPAR5 = 172352	TST103 013654	TST166 032122
ADDW8 = 000000	BIT12 = 010000	KPAR6 = 172354	TST104 014432	TST167 032326
ADDW9 = 000000	BIT13 = 020000	KPAR7 = 172356	TST105 014644	TST17 004066
ADEVC1 = 000000	BIT14 = 040000	KPDR0 = 172300	TST106 014776	TST170 032430
ADEVM = 000000	BIT15 = 100000	KPDR1 = 172302	TST107 015134	TST171 032630
ADRJMP 002134	CA121 050506	KPDR2 = 172304	TST11 003426	TST172 033166
ADRSYN 002132	CA210 050464	KPDR3 = 172306	TST110 015320	TST173 033524
ADRIS 002136	CA2113 050476	KPDR4 = 172310	TST111 015516	TST174 034014
AENV = 000000	CCR = 177746	KPDR5 = 172312	TST112 015716	TST175 034304
AENVM = 000000	CDR = 177754	KPDR6 = 172314	TST113 016114	TST176 034662
AFATAL = 000000	CDR150 050502	KPDR7 = 172316	TST114 016314	TST177 035314
AM 000010	CHR = 177752	LF = 000012	TST115 016536	TST2 002672
AMADR1 = 000000	CHRPAT 002066	LOOP 002062	TST116 016732	TST20 004136
AMADR2 = 000000	CHR157 050474	LOW1 060000	TST117 017324	TST200 035746
AMADR3 = 000000	CHR50 050524	LPB = 002000	TST12 003506	TST201 036302
AMADR4 = 000000	CHR80 050500	MAGPRE 007074	TST120 017420	TST202 036636
AMAMS1 = 000000	CME 177744	OFF = 001015	TST121 017476	TST203 037172
AMAMS2 = 000000	CMPE 100000	PEA = 000200	TST122 017604	TST204 037526
AMAMS3 = 000000	CMR = 177750	PEHI = 000200	TST123 017712	TST205 037650
AMAMS4 = 000000	CMRPAT 002064	PELO = 000100	TST124 020130	TST206 037772
AMR210 050470	CMR119 050526	PRDATA 050064	TST125 020324	TST207 040114
AMSGAD = 000000	CM1 = 100000	PRTABL 050420	TST126 020542	TST21 004216
AMSGLG = 000000	CM1513 050512	PRTITL 047056	TST127 020744	TST210 040300
AMSGTY = 000000	CM2 = 040000	PSW = 177776	TST13 003602	TST211 040464
AMTYP1 = 000000	CM3 = 020000	RDCHR = 104407	TST130 021212	TST212 040606
AMTYP2 = 000000	CNT121 050514	RDDEC = 104412	TST131 021470	TST213 040736
AMTYP3 = 000000	CPSAVE 046220	RDLIN = 104410	TST132 021746	TST214 041104
AMTYP4 = 000000	CR 000015	RDOCT = 104411	TST133 022164	TST215 041316
APASS = 000000	CRLF 000200	RECDAT 050520	TST134 022366	TST216 041434
APRIOR = 000000	DCPI = 000001	RELCTH 002452	TST135 022604	TST217 041552
APTCSU = 000040	DISPRE 000174	RELCTL 002424	TST136 023006	TST22 004262
APTENV = 000001	EHA = 000004	SAVR0 046204	TST137 023254	TST220 042034
APTSIZ = 000200	ENDPAS 044076	SAVR1 046206	TST14 003652	TST221 042202
APTSPO = 000100	ERROR = 104413	SAVR2 046210	TST140 023532	TST222 042732
ASWREG = 000000	ERRRPC 046202	SAVR3 046212	TST141 024010	TST223 043326
ATESTN = 000000	ESA = 000020	SAVR4 046214	TST142 024234	TST224 043472
AUNIT = 000000	EXDAT1 050510	SAVR5 046216	TST143 024444	TST225 043640
AUSWR = 000000	EXDAT3 050522	SLOPE = 000004	TST144 024644	TST226 043774
AVECT1 = 000000	EXDAT6 050504	SCPCND 000004	TST145 025116	TST23 004332
AVECT2 = 000000	FAILAD 050530	SRO = 177572	TST146 025300	TST24 004402

TST25	004452	TST7	003322	\$CDW1	001546	\$FILLC	002110	\$RDOCT	045562
TST26	004522	TST70	010556	\$CDW2	001550	\$FILLS	002107	\$RDSZ =	000010
TST27	004572	TST71	010724	\$CHARC	044602	\$HD =	000003	\$SCPSE	002144
*ST3	002766	TST72	011254	\$CNTLG	045533	\$HIBTS	001452	\$SETUP=	000000
TST30	004642	TST73	011442	\$CNTLU	045526	\$HIOCT	045662	\$SWR =	160000
TST31	004712	TST74	011634	\$CPUOP	001514	\$ICNT	044240	\$SWREG	001510
TST32	004762	TST75	012044	\$CRLF	002117	\$LF	002120	\$TESTN	001472
TST33	005032	TST76	012264	\$DBLK	045250	\$LFLG	002055	\$TIMES	044236
TST34	005102	TST77	012510	\$DDW0	001552	\$MADR1	001520	\$TKB	002100
TST35	005152	TYPE =	104406	\$DDW1	001554	\$MADR2	001524	\$TKS	002076
*ST36	005222	YPDS =	104405	\$DDW10	001576	\$MADR3	001530	STN =	000227
TST37	005272	TYPE =	104401	\$DDW11	001600	\$MADR4	001534	\$TPB	002104
TST4	003062	TYPC =	104402	\$DDW12	001602	\$MAIL	001466	\$TPFLG	002111
*ST40	005342	TYPON =	104404	\$DDW13	001604	\$MAMS1	001516	\$TPS	002102
TST41	005412	TYPOS =	104403	\$DDW14	001606	\$MAMS2	001522	\$TRAP	046116
*ST42	005462	UCB =	001000	\$DDW15	001610	\$MAMS3	001526	\$TRAP2	046140
TST43	005532	UMPRO0=	170200	\$DDW2	001556	\$MAMS4	001532	\$TRP =	000014
TST44	005602	UMPRO1=	170202	\$DDW3	001560	\$MBADR	001454	\$TRPAD	046152
TST45	005652	UMPRO2=	170204	\$DDW4	001562	\$MFLG	002054	\$STIM	001456
TST46	005722	UMPRO3=	170206	\$DDW5	001564	\$MNEW	045551	\$STSTM	002060
TST47	005772	UMPRO4=	170210	\$DDW6	001566	\$MSGAD	001502	\$TTYIN	045516
TST5	003156	UMPRO5=	170212	\$DDW7	001570	\$MSGLG	001504	\$TYPBN	046042
TST50	006050	UMPRO6=	170214	\$DDW8	001572	\$MSGTY	001466	\$TYPDS	045034
TST51	006126	UMPRO7=	170216	\$DDW9	001574	\$MSWR	045540	\$TYPE	044242
TST52	006204	UMPRO8=	170220	\$DEVCT	001476	\$MTYP1	001517	\$TYPEC	044454
TST53	006262	UMPRO9=	170222	\$DEVM	001544	\$MTYP2	001523	\$TYPEX	044604
TST54	006340	VCIP =	010000	\$DOAGN	044232	\$MTYP3	001527	\$TYPC	044632
TST55	006416	VLD	010000	\$DTBL	045240	\$MTYP4	001533	\$TYPON	044646
TST56	006506	VSU	020000	\$ENDAD	044222	\$NULL	002106	\$TYPOS	044606
TST57	006604	WVPD =	000100	\$ENULL	002122	\$NWTST=	000001	\$UNIT	001500
TST6	003252	WVPT =	002000	\$ENV	001506	\$OCNT	045030	\$UNITM	001462
TST60	006654	\$APTHD	001452	\$ENVM	001507	\$OMODE	045032	\$USWR	001512
*ST61	006724	\$ATYC	001636	\$EROVR	046230	\$PASS	001474	\$VECT1	001536
TST62	006774	\$ATY1	001612	\$ERROR	046224	\$PASTM	001460	\$VECT2	001540
TST63	007270	\$ATY3	001620	\$ERRPC	002422	\$QUES	002116	\$XOFF =	000023
TST64	007356	\$ATY4	001630	\$ETABL	001506	\$RDCHR	045260	\$XON =	000021
TST65	007554	\$BASE	001542	\$ETEND	001612	\$RDDEC	045664	\$OFILL	045031
TST66	007734	\$BELL	002112	\$FATAL	001470	\$RDLIN	045410	.\$X =	001452
TST67	010200	\$BIN	046114	\$FFLG	002056				

. ABS. 070002 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 32328 WORDS (127 PAGES)
DYNAMIC MEMORY: 20034 WORDS (77 PAGES)
ELAPSED TIME: 00:06:57
CKKAC.BIN,CKKAC.SEQ/CR/NL:TOC/-SP=CKKAC.MLB/ML,CKKAC.P11

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ABASE	=	000000	16-577 16-577
ADDW1	=	000000	16-577 16-577
ADDW2	=	000000	16-577 16-577
ACPUOP	=	000000	16-577 16-577
ADDW0	=	000000	16-577 16-577
ADDW1	=	000000	16-577 16-577
ADDW10	=	000000	16-577 16-577
ADDW11	=	000000	16-577 16-577
ADDW12	=	000000	16-577 16-577
ADDW13	=	000000	16-577 16-577
ADDW14	=	000000	16-577 16-577
ADDW15	=	000000	16-577 16-577
ADDW2	=	000000	16-577 16-577
ADDW3	=	000000	16-577 16-577
ADDW4	=	000000	16-577 16-577
ADDW5	=	000000	16-577 16-577
ADDW6	=	000000	16-577 16-577
ADDW7	=	000000	16-577 16-577
ADDW8	=	000000	16-577 16-577
ADDW9	=	000000	16-577 16-577
ADEVCT	=	000000	16-577 16-577
ADEVMI	=	000000	16-577 16-577
ADR JMP		002134	#19-728 *19-771 19-773 19-775 182-7334 182-7337
ADRSYN		002132	#19-727 *19-770 182-7339
ADR1\$		002136	#19-729 *19-773 *19-774 19-776 182-7335 182-7338
AENV	=	000000	16-577 16-577
AENVMI	=	000000	16-577 16-577
AFATAL	=	000000	16-577 16-577
AM		000010	#18-698 68-1489 72-1560 73-1612 74-1644 75-1683 76-1735 77-1803 78-1852
AMADR1	=	000000	79-1903 16-577
AMADR2	=	000000	16-577 16-577
AMADR3	=	000000	16-577 16-577
AMADR4	=	000000	16-577 16-577
AMAMS1	=	000000	16-577 16-577
AMAMS2	=	000000	16-577 16-577
AMAMS3	=	000000	16-577 16-577
AMAMS4	=	000000	16-577 16-577
AMR210		050470	*74-1648 *74-1650 *74-1652 *74-1653 74-1661 *75-1687 *75-1688 *75-1689 75-1695
			*76-1737 *76-1739 76-1746 *77-1807 *77-1809 *77-1813 77-1820 *78-1854 *78-1856
			78-1865 *79-1907 *79-1910 *79-1913 *79-1915 79-1923 182-7396 182-7397 183-7432
			#183-7436
AMSGAD	=	000000	16-577 16-577
AMSGLG	=	000000	16-577 16-577
AMSGTY	=	000000	16-577 16-577
AMTYP1	=	000000	16-577 16-577
AMTYP2	=	000000	16-577 16-577
AMTYP3	=	000000	16-577 16-577
AMTYP4	=	000000	16-577 16-577
APASS	=	000000	16-577 16-577
APRIOR	=	000000	16-577
APTC SU		000040	#17-579 #18-680 173-7216

SYMBOL CROSS REFERENCE		REFERENCES		CREF	V01					
SYMBOL	VALUE									
APTENV	= 000001	17-579	#17-579	#18-678	173-7216	182-7321				
APTSIZ	= 000200	14-563	#17-579	#18-677	89-2340	167-6911	168-6999			
APTSPO	= 000100	17-579	#17-579	#18-679	173-7216					
ASWREG	= 000000	16-577	16-577							
ATESTN	= 000000	16-577	16-577							
AUNJY	= 000000	16-577	16-577							
AUSWR	= 000000	16-577	16-577							
AVECT1	= 000000	16-577	16-577							
AVECT2	= 000000	16-577	16-577							
BABA	= 170004	#18-672	*167-6945	*168-7019						
BECC	= 170002	#18-671	*167-6947	*168-7020						
BEFR1	= 170006	#18-674	167-6918	*167-6950	167-6951	168-7006	*168-7023	168-7024		
BECR2	= 170016	#18-675	*167-6949	*168-7022	168-7029					
BEDA	= 170000	#18-673	*167-6948	*168-7021						
BEGIN	002500	14-574	#21-802	172-7212						
BIT00	000001	#18-625	19-736	19-738	31-978	31-981	49-1251	49-1253	63-1409	63-1413
		64-1424	64-1428	182-7262	182-7264					
BIT01	= 000002	#18-626	42-1146	42-1148	50-1261	50-1263	65-1439	65-1443	66-1454	66-1458
BIT02	000004	#18-627	32-994	32-998	33-1013	51-1271	51-1273	61-1376	61-1377	61-1379
		62-1393	62-1394							
BIT03	= 000010	#18-628	34-1026	34-1030	35-1045	52-1281	52-1283	67-1475	68-1493	
BIT04	= 000020	#18-629	43-1161	43-1163	53-1291	53-1293				
BIT05	= 000040	#18-630	44-1176	44-1178	69-1505	69-1507				
BIT06	000100	#18-631	36-1058	36-1060	70-1515	70-1517				
BIT07	= 000200	#18-632	37-1073	37-1075	38-1088	38-1092	71-1525	71-1527		
BIT08	= 000400	#18-633	31-977	39-1105	39-1107	45-1191	45-1193	54-1301	54-1303	122-3888
BIT09	= 001000	#18-634	40-1121	55-1311	55-1313	182-7329	182-7348			
BIT10	= 002000	#18-635	41-1132	41-1134	56-1321	56-1323				
BIT11	= 004000	#18-636	19-761	46-1206	46-1208	57-1331	57-1333	172-7192		
BIT12	= 010000	#18-637	30-961	58-1341	58-1343					
BIT13	= 020000	#18-638	59-1351	59-1353	182-7253					
BIT14	= 040000	#18-639	19-750	47-1221	47-1223	60-1361	60-1363			
BIT15	= 100000	#18-640	48-1236	48-1238	182-7325					
CA121	050506	*90-2454	*90-2455	90-2460	*98-2749	*98-2750	98-2755	*105-3012	*105-3014	105-3019
		*106-3057	*106-3059	106-3064	*107-3107	*107-3109	107-3114	*108-3151	*108-3153	108-3158
		*109-3204	*109-3205	109-3212	*112-3372	*112-3374	112-3379	*113-3417	*113-3419	113-3424
		*114-3468	*114-3470	114-3475	*115-3512	*115-3514	115-3519	*116-3565	*116-3566	116-3573
		*124-4004	*124-4005	124-4019	*124-4029	*124-4030	124-4042	*125-4100	*125-4101	125-4116
		*125-4126	*125-4127	125-4139	*126-4200	*126-4201	126-4216	*126-4231	*126-4232	126-4245
		*127-4306	*127-4307	127-4322	*127-4337	*127-4338	127-4351	*135-4741	*135-4742	135-4750
		135-4760	135-4768	*136-4813	*136-4814	136-4822	136-4832	136-4840	*137-4884	*137-4885
		*137-4893	137-4903	137-4911	*138-4955	*138-4956	138-4964	138-4974	138-4982	*139-5038
		*139-5039	139-5047	*139-5057	*139-5058	139-5066	*140-5117	*140-5118	140-5126	*148-5686
		*148-5687	148-5702	*148-5718	*148-5719	148-5730	*149-5811	*149-5812	149-5827	*149-5843
		*149-5844	149-5856	*150-5921	*150-5922	150-5934	*150-5942	*150-5943	150-5951	*151-6016
		*151-6017	151-6027	*151-6035	*151-6036	151-6044	*152-6109	*152-6110	152-6122	*152-6130
		*152-6131	152-6140	*153-6207	*153-6208	153-6220	*153-6228	*153-6229	153-6237	*167-6968
		*167-6970	167-6977	182-7409	183-7432	#183-7442				
CA210	050464	*74-1646	*74-1647	74-1660	*75-1685	*75-1686	75-1694	*76-1740	*76-1741	76-1745
		*77-1808	*77-1810	*77-1812	77-1819	*78-1857	*78-1858	78-1863	*79-1909	*79-1911
		*79-1914	*79-1916	79-1921	*83-2070	*83-2071	83-2082	*84-2117	*84-2118	84-2128
		*85-2162	*85-2163	85-2174	*86-2209	*86-2210	86-2220	*94-2586	*94-2587	94-2592

SYMBOL CROSS REFERENCE
SYMBOL VALUE

REFERENCES
CREF V01

SYMBOL	CROSS REFERENCE VALUE	REFERENCES	CREF	V01						
		*95-2624	*95-2625	95-2630	*96-2662	*96-2663	96-2668	*97-2700	*97-2701	97-2706
		182-7393	182-7394	183-7432	#183-7435					
A 113	050476	*88-2305	*88-2308	*88-2310	88-2319	*89-2386	*89-2389	*89-2391	89-2400	*100-2866
		*100-2868	*100-2869	100-2876	182-7401	183-7432	#183-7438			
R	*77746	*14-547	#18-612	*21-805	24-845	29-948	30-961	*31-977	*31-978	31-979
		*32-993	*32-994	32-996	*32-997	*33-1011	33-1013	*34-1025	*34-1026	34-1027
		*34-1029	*35-1043	35-1045	*36-1058	36-1060	*37-1073	37-1075	*38-1088	38-1089
		*38-1091	*39-1105	39-1107	*40-1119	40-1121	*41-1132	41-1134	*42-1146	42-1148
		*43-1161	43-1163	*44-1176	44-1178	*45-1191	45-1193	*46-1206	46-1208	*47-1221
		47-1223	*48-1236	48-1238	*61-1376	*61-1377	61-1379	*62-1393	*62-1394	62-1396
		*74-1627	*75-1673	*76-1708	*77-1761	*78-1838	*79-1876	*80-1936	*80-1944	*80-1952
		*81-1971	*81-1979	*81-1987	*82-2006	*82-2008	*82-2024	*83-2048	*83-2052	*83-2066
		*84-2095	*84-2099	*84-2113	*85-2140	*85-2144	*85-2158	*86-2187	*86-2191	*86-2205
		*87-2233	*87-2238	*87-2247	*88-2270	*88-2284	*88-2298	*89-2339	*89-2366	*89-2379
		*90-2427	*90-2438	*90-2450	*91-2473	*91-2478	*91-2488	*92-2502	*92-2507	*92-2517
		*93-2531	*93-2537	*93-2547	*94-2566	*94-2576	*94-2583	*95-2604	*95-2614	*95-2621
		*96-2642	*96-2652	*96-2659	*97-2680	*97-2690	*97-2697	*98-2717	*98-2732	*98-2745
		*99-2772	*99-2777	*99-2787	*100-2812	*100-2838	*100-2856	*100-2890	*101-2901	101-2902
		*102-2921	102-2922	103-2938	*103-2940	103-2941	*103-2943	103-2944	103-2947	104-2960
		*104-2962	104-2963	*104-2965	104-2966	104-2969	*105-2982	105-2983	*105-2985	105-2986
		*105-2993	*105-3009	106-3033	*106-3035	106-3036	*106-3040	*106-3053	*107-3077	107-3078
		*107-3080	107-3081	*107-3088	*107-3104	*108-3127	108-3128	*108-3130	108-3131	*108-3135
		*108-3148	*109-3169	109-3170	*109-3172	109-3173	*109-3183	*109-3200	*110-3226	110-3229
		*110-3231	110-3232	*110-3234	*110-3245	110-3247	*110-3249	110-3250	*110-3261	*111-3284
		111-3287	*111-3289	111-3290	*111-3292	*111-3303	111-3305	*111-3307	111-3308	*111-3319
		*112-3342	112-3343	*112-3345	112-3346	*112-3353	*112-3369	*113-3392	113-3393	*113-3395
		113-3396	*113-3400	*113-3413	*114-3437	114-3438	*114-3440	114-3441	*114-3448	*114-3464
		*115-3488	115-3489	*115-3491	115-3492	*115-3496	*115-3509	*116-3530	116-3531	*116-3533
		116-3534	*116-3544	*116-3561	*117-3587	117-3590	*117-3592	117-3593	*117-3595	*117-3606
		117-3608	*117-3610	117-3611	*117-3622	*118-3645	118-3648	*118-3650	118-3651	*118-3653
		*118-3664	118-3666	*118-3668	118-3669	*118-3680	*119-3703	*119-3711	*119-3729	*120-3763
		*120-3769	*120-3781	*121-3808	*121-3813	*121-3828	*122-3859	122-3860	*122-3862	122-3863
		*122-3865	*122-3873	*122-3884	*123-3924	123-3925	*123-3927	123-3928	*123-3932	*123-3940
		*123-3945	*124-3970	*124-3982	*124-4000	*124-4024	*124-4048	*125-4065	*125-4077	*125-4095
		*125-4121	*125-4145	*126-4162	*126-4170	*126-4190	*126-4222	*126-4251	*127-4268	*127-4276
		*127-4296	*127-4328	*127-4357	*128-4375	*128-4381	*128-4390	*129-4423	*129-4430	*129-4441
		*130-4481	*130-4488	*130-4497	*131-4532	*131-4538	*131-4548	*131-4550	131-4552	*132-4586
		*132-4594	*132-4603	*133-4629	*133-4639	*133-4648	*134-4673	*134-4681	*134-4690	*134-4692
		134-4694	*135-4718	*135-4732	*135-4740	*136-4789	*136-4803	*136-4812	*137-4861	*137-4875
		*137-4883	*138-4932	*138-4946	*138-4954	*139-5002	*139-5019	*139-5033	*140-5089	*140-5099
		*140-5113	*141-5140	*142-5169	*142-5170	142-5171	*142-5177	*142-5186	*142-5192	*143-5229
		*143-5230	143-5231	*143-5237	*143-5240	*143-5245	*143-5253	*143-5261	*143-5264	143-5266
		*144-5329	*144-5330	144-5331	*144-5337	*144-5340	*144-5346	*144-5355	*144-5364	*144-5367
		144-5369	*145-5413	*145-5414	145-5415	*145-5424	*145-5427	*145-5431	*145-5433	*145-5442
		*145-5444	*145-5451	145-5453	*146-5469	*146-5470	146-5471	*146-5481	*146-5484	*146-5488
		*146-5490	*146-5497	*146-5499	*146-5507	*146-5509	*147-5533	*147-5534	147-5535	*147-5547
		*147-5550	*147-5554	*147-5556	*147-5557	*147-5571	*147-5573	*147-5583	147-5585	*148-5629
		*148-5644	*148-5676	*148-5708	*148-5736	*149-5753	*149-5768	*149-5801	*149-5833	*149-5862
		*150-5879	150-5880	*150-5882	150-5883	*150-5893	*150-5900	*150-5905	*150-5919	*150-5940
		*150-5957	*151-5974	151-5975	*151-5977	151-5978	*151-5987	*151-5994	*151-5999	*151-6014
		*151-6033	*151-6050	*152-6067	152-6068	*152-6070	152-6071	*152-6080	*152-6087	*152-6092
		*152-6107	*152-6128	*152-6146	*153-6163	153-6164	*153-6166	*153-6167	*153-6177	*153-6184

SYMBOL CROSS REFERENCE
 SYMBOL VALUE

REFERENCES
 CREF V01

SYMBOL	CROSS REFERENCE VALUE	REFERENCES	CREF	V01						
		*153-6189	*153-6205	*153-6226	*153-6243	*154-6266	*154-6272	*154-6281	*155-6309	*155-6315
		*155-6325	*156-6353	*156-6359	*156-6368	*157-6393	*157-6398	*157-6401	*157-6406	*157-6415
		*157-6417	157-6418	*158-6444	*158-6449	*158-6452	*158-6458	*158-6469	*158-6471	158-6472
		*159-6501	*159-6507	*159-6515	*160-6543	*160-6549	*160-6553	*160-6557	*161-6591	*161-6599
		*161-6604	*161-6608	*161-6615	*162-6644	*162-6658	*162-6676	*163-6707	*163-6712	*163-6724
		*164-6754	*164-6759	*164-6772	*165-6787	*165-6803	*165-6836	*166-6863	*166-6869	*166-6881
		*167-6903	167-6904	*167-6906	167-6907	*167-6934	*167-6964	*168-6991	168-6992	*168-6994
		168-6995	*168-7015	168-7030	*168-7039	*169-7061	*169-7067	*169-7071	*170-7093	*170-7099
		*170-7109	*171-7140							
CDR	= 177754	#18-615	27-909	91-2485	92-2514	93-2544	94-2580	95-2618	96-2656	97-2694
		98-2742	119-3720	119-3726	124-3991	124-3996	125-4086	125-4091		
CDR150	050502	*91-2485	91-2489	91-2495	*92-2514	92-2518	92-2524	*93-2544	93-2548	93-2556
		*94-2580	94-2584	94-2593	*95-2618	95-2622	95-2631	*96-2656	96-2660	96-2669
		*97-2694	97-2698	97-2707	*119-3734	119-3741	*119-3746	119-3753	*124-4003	124-4017
		*124-4027	124-4040	*125-4098	125-4114	*125-4124	125-4137	182-7405	183-7432	#183-7440
CHR	- 177752	#18-614	26-888	*67-1469	*68-1486	*72-1541	*72-1544	*73-1606	*74-1629	*75-1676
		*76-1719	*77-1779	*78-1840	*79-1889	*80-1937	80-1949	*81-1972	81-1984	*82-2009
		82-2021	*83-2049	83-2063	*84-2096	84-2110	*85-2141	85-2155	*86-2188	86-2202
		87-2244	88-2294	89-2374	90-2447	*105-2988	*107-3083	*112-3348	*114-3443	*120-3764
		120-3780	121-3825	*126-4163	*126-4175	126-4182	126-4185	*127-4269	*127-4281	127-4288
		127-4291	*132-4587	*133-4631	*134-4674	*135-4719	*136-4790	*137-4862	*138-4933	*148-5631
		*148-5650	*149-5755	*149-5774	169-7075	170-7107	*182-7339			
CHRPAT	002066	#18-588	*74-1628	74-1629	74-1648	74-1650	*74-1663	*77-1774	77-1779	77-1807
		77-1808	*77-1827	*79-1878	79-1889	*79-1928				
CHR157	050474	*80-1949	*80-1953	80-1954	*80-1957	*80-1958	80-1965	*81-1984	*81-1988	81-1989
		*81-1992	*81-1993	81-2000	*82-2021	*82-2026	*82-2027	82-2030	82-2037	*83-2063
		*83-2067	83-2068	*83-2073	*83-2074	83-2081	*84-2110	*84-2114	84-2115	*84-2120
		*84-2121	84-2129	*85-2155	*85-2159	85-2160	*85-2165	*85-2166	85-2173	*86-2202
		*86-2206	86-2207	*86-2212	*86-2213	86-2221	*87-2244	*87-2248	87-2249	*87-2252
		*87-2253	87-2260	*88-2294	*88-2300	88-2301	*88-2307	*88-2309	88-2318	*89-2374
		*89-2381	89-2382	*89-2388	*89-2390	89-2399	*120-3780	*120-3784	120-3785	*120-3788
		*120-3789	120-3801	*121-3825	*121-3829	121-3830	*121-3833	*121-3834	121-3843	*126-4193
		*126-4196	*126-4197	126-4214	*126-4225	*126-4227	*126-4228	126-4244	*127-4299	*127-4302
		*127-4303	127-4320	*127-4331	*127-4333	*127-4334	127-4350	182-7399	183-7432	#183-7437
CHR50	050524	*169-7082	169-7086	*170-7114	170-7118	182-7423	183-7433	#183-7449		
CHR80	050500	*82-2007	82-2009	82-2030	82-2038	*82-2041	82-2042	182-7403	183-7432	#183-7439
CME	- 177744	#18-611	*19-749	23-823	*28-931	28-933	*49-1251	49-1253	*50-1261	50-1263
		*51-1271	51-1273	*52-1281	52-1283	*53-1291	53-1293	*54-1301	54-1303	*55-1311
		55-1313	*56-1321	56-1323	*57-1331	57-1333	*58-1341	58-1343	*59-1351	59-1353
		*60-1361	60-1363	*141-5146	141-5147	*142-5182	142-5190	*143-5246	143-5252	143-5256
		*143-5258	143-5259	*144-5347	144-5354	144-5358	*144-5361	144-5362	*145-5432	*146-5489
		*147-5555	147-5575	*147-5576	147-5577	*157-6407	*158-6459			
CMPE	- 100000	#18-713	147-5587	147-5594						
CMR	177750	#18-613	25-867	*63-1409	63-1410	*63-1412	*64-1424	64-1425	*64-1427	*65-1439
		65-1440	*65-1442	*66-1454	66-1455	*66-1457	*67-1470	*67-1471	67-1472	*67-1474
		*68-1487	*68-1488	*68-1489	68-1490	*68-1492	*69-1505	69-1507	*70-1515	70-1517
		*71-1525	71-1527	*72-1542	*72-1543	*72-1545	72-1558	*73-1607	*73-1608	73-1612
		*74-1630	*74-1631	74-1642	*75-1675	*75-1677	75-1681	*76-1720	*76-1721	76-1731
		*77-1778	*77-1782	77-1799	*78-1841	*78-1842	78-1852	*79-1888	*79-1892	79-1903
		*80-1938	*80-1951	*81-1973	*81-1986	*82-2011	*82-2023	*83-2053	*83-2065	*84-2100
		*84-2112	*85-2145	*85-2157	*86-2192	*86-2204	*87-2234	*87-2246	*88-2279	*88-2299
		*89-2363	*89-2380	*90-2435	*90-2449	*91-2475	*91-2487	*92-2504	*92-2516	*93-2534

SYMBOL CROSS REFERENCE
 SYMBOL VALUE

REFERENCES

SYMBOL	CROSS REFERENCE VALUE	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	
		*93-2546	*94-2573	*94-2582	*95-2611	*95-2620	*96-2649	*96-2658	*97-2687	*97-2696
		*98-2729	*98-2744	*99-2773	99-2784	*99-2786	*100-2835	100-2852	*100-2857	*100-2889
		*105-2990	*105-2994	105-3006	*105-3008	*106-3041	106-3050	*106-3052	*107-3085	*107-3089
		107-3101	*107-3103	*108-3136	108-3145	*108-3147	*109-3180	*109-3184	*109-3189	109-3197
		*109-3199	*110-3235	110-3257	*110-3260	*111-3293	111-3315	*111-3318	*112-3350	*112-3354
		112-3366	*112-3368	*113-3401	113-3410	*113-3412	*114-3445	*114-3449	114-3461	*114-3463
		*115-3497	115-3506	*115-3508	*116-3541	*116-3545	*116-3550	116-3558	*116-3560	*117-3596
		117-3618	*117-3621	*118-3654	118-3676	*118-3679	*119-3708	*119-3728	*120-3765	*120-3772
		*120-3783	*121-3809	*121-3827	*122-3866	122-3879	122-3883	*122-3886	*123-3933	123-3944
		*123-3947	*124-3978	*124-4001	*124-4025	*124-4049	*125-4073	*125-4096	*125-4122	*125-4146
		*126-4166	*126-4191	*126-4223	*126-4252	*127-4272	*127-4297	*127-4329	*127-4358	*128-4377
		128-4387	*128-4389	*129-4426	129-4438	*129-4440	*130-4484	130-4494	*130-4496	*131-4534
		131-4546	*131-4549	*132-4588	132-4600	*132-4602	*133-4633	133-4645	*133-4647	*134-4675
		134-4688	*134-4691	*135-4726	135-4737	*135-4739	*136-4797	136-4809	*136-4811	*137-4869
		137-4880	*137-4882	*138-4940	138-4951	*138-4953	*139-5015	139-5030	*139-5032	*140-5095
		140-5110	*140-5112	*141-5141	*141-5149	*142-5174	*142-5193	*143-5234	*143-5262	*144-5334
		*144-5365	*145-5420	*145-5447	*146-5476	*146-5503	*147-5541	*147-5579	*148-5638	*148-5652
		148-5660	*148-5661	148-5671	*148-5677	*148-5709	*148-5737	*149-5762	*149-5776	*149-5784
		*149-5785	149-5796	*149-5802	*149-5834	*149-5863	*150-5888	150-5912	150-5914	*150-5920
		*150-5941	*150-5958	*151-5983	151-6006	151-6009	*151-6015	*151-6034	*151-6051	*152-6076
		152-6099	152-6102	*152-6108	*152-6129	*152-6147	*153-6172	153-6197	153-6200	*153-6206
		*153-6227	*153-6244	*154-6268	154-6279	*155-6311	155-6323	*156-6355	156-6366	*157-6395
		157-6414	*158-6446	158-6468	*159-6503	159-6513	*160-6545	160-6556	*161-6595	161-6613
		*162-6654	162-6669	162-6671	*163-6708	163-6722	*164-6755	164-6770	*165-6798	*165-6819
		*165-6835	*167-6935	*167-6942	*167-6953	167-6962	*167-6965	*169-7064	*169-7078	*170-7096
		*170-7110	*182-7340							
CMR PAT	002064	*18-587	*75-1674	75-1675	75-1688	*75-1697	75-1698	*77-1773	77-1778	*77-1826
CMR119	050526	*79-1877	79-1888	*79-1927						
		*148-5679	*148-5682	*148-5683	148-5700	*148-5711	*148-5714	*148-5715	148-5728	*149-5804
		*149-5807	*149-5808	149-5825	*149-5836	*149-5839	*149-5840	149-5854	182-7425	183-7433
		*183-7450								
CM1	= 100000	*18-707								
CM1513	050512	*99-2784	*99-2788	99-2789	*99-2793	*99-2794	99-2802	*100-2852	*100-2859	*100-2860
		100-2863	100-2879	182-7413	183-7432	*183-7444				
CM2	= 040000	*18-706								
CM3	= 020000	*18-705								
CNT121	050514	*110-3262	*110-3264	110-3271	*111-3320	*111-3322	111-3329	*117-3623	*117-3625	117-3632
		*118-3681	*118-3683	118-3690	182-7415	183-7433	*183-7445			
LPSAVE	046220	*19-735	19-736	19-743	*181-7242	182-7257	*182-7261	182-7262	182-7269	182-7429
		183-7433								
CR	- 000015	173-7216	173-7216							
CRLF	- 000200	14-553	14-554	14-555	14-556	14-572	89-2416	89-2417	167-6930	167-6930
		167-6932	167-6932	172-7202	173-7216	173-7216	182-7272	182-7272	182-7283	
DCPI	- 000001	*18-683	145-5433							
DISPRE	000174	*13-518								
EHA	= 000004	*18-697								
ENDPAS	044076	*172-7190								
ERROR	= '04413	19-741	23-831	24-853	25-875	26-896	27-917	28-935	29-951	30-964
		31-983	32-1000	33-1015	34-1032	35-1047	36-1062	37-1077	38-1094	40-1123
		41-1136	42-1150	43-1165	44-1180	45-1195	46-1210	47-1225	48-1240	49-1255
		50-1265	51-1275	52-1285	53-1295	54-1305	55-1315	56-1325	57-1335	58-1345
		59-1355	60-1365	61-1381	62-1398	63-1415	64-1430	65-1445	66-1460	67-1477

SYMBOL CROSS REFERENCE SYMBOL VALUE

REFERENCES CREF V01

SYMBOL	CROSS REFERENCE VALUE	REFERENCES	CREF	V01						
		68-1495	69-1509	70-1519	71-1529	72-1562	73-1614	74-1656	75-1690	76-1742
		77-1816	78-1859	79-1919	80-1961	81-1996	82-2032	83-2077	84-2124	85-2169
		86-2216	87-2256	88-2313	89-2394	90-2456	91-2491	92-2520	93-2551	94-2588
		95-2626	96-2664	97-2702	98-2751	99-2797	100-2873	101-2910	102-2927	103-2950
		104-2972	105-3015	106-3060	107-3110	108-3154	109-3206	110-3265	111-3323	112-3375
		113-3420	114-3471	115-3515	116-3567	117-3626	118-3684	119-3735	119-3747	120-3793
		121-3838	122-3892	122-3900	122-3909	122-3917	123-3950	124-4006	124-4031	125-4102
		125-4128	126-4202	126-4233	127-4308	127-4339	128-4393	128-4401	129-4444	129-4454
		130-4500	130-4510	131-4556	131-4564	132-4606	133-4651	134-4698	135-4745	135-4755
		135-4764	136-4817	136-4827	136-4836	137-4888	137-4898	137-4907	138-4959	138-4969
		138-4978	139-5040	139-5059	140-5119	141-5153	142-5199	143-5273	143-5284	143-5295
		144-5376	144-5387	144-5398	145-5457	146-5513	147-5589	147-5596	147-5603	147-5610
		148-5688	148-5720	149-5813	149-5845	150-5923	150-5944	151-6018	151-6037	152-6111
		152-6132	153-6209	153-6230	154-6284	155-6328	156-6371	157-6422	158-6476	159-6518
		160-6561	161-6618	162-6681	163-6727	164-6775	165-6842	165-6852	166-6886	167-6971
		168-7042	168-7051	169-7083	170-7115	171-7144	172-7182	#180-7231		
ERRPC	046202	#181-7235	*182-7256	182-7271	182-7286					
ESA	= 000020	#18-699								
EXDAT1	050510	*99-2791	99-2801	*100-2872	100-2878	*148-5678	148-5699	*148-5710	148-5727	*149-5803
		149-5824	*149-5835	149-5853	182-7411	183-7432	#183-7443			
EXDAT3	050522	*120-3792	120-3800	*121-3837	121-3842	*126-4192	126-4213	*126-4224	126-4241	*127-4298
		127-4319	*127-4330	127-4347	182-7421	183-7433	#183-7448			
EXDAT6	050504	*93-2550	93-2555	*119-3733	119-3740	*119-3745	119-3752	*124-4002	124-4016	*124-4026
		124-4039	*125-4097	125-4113	*125-4123	125-4136	*142-5197	142-5202	*143-5271	143-5277
		*143-5282	143-5288	*143-5293	143-5299	*144-5374	144-5380	*144-5385	144-5391	*144-5396
		144-5402	*166-6885	166-6891	*171-7142	171-7146	182-7407	183-7432	#183-7441	
FAILAD	050530	*171-7141	171-7145	182-7427	183-7433	#183-7451				
FAIL1	002070	#18-589	*101-2900	*101-2904	101-2908	*145-5423	*145-5441	145-5455	*146-5480	*146-5500
		146-5511	*147-5545	*147-5570	147-5608	*165-6792	*165-6831	165-6840		
FAIL2	002072	#18-590	*165-6793	*165-6833	165-6849					
FC	- 000400	#18-688	101-2901	102-2921	103-2940	103-2943	104-2962	104-2965	105-2985	106-3035
		107-3080	108-3130	109-3172	110-3231	110-3245	110-3249	111-3289	111-3303	111-3307
		112-3345	113-3395	114-3440	115-3491	116-3533	117-3592	117-3606	117-3610	118-3650
		118-3664	118-3668	122-3862	123-3927	131-4550	134-4692	142-5170	143-5230	143-5264
		144-5330	144-5367	145-5414	145-5451	146-5470	146-5507	147-5534	147-5583	150-5882
		151-5977	152-6070	153-6166	157-6417	158-6471	167-6906	168-6994		
FLTPAT	050516	*76-1709	76-1719	76-1722	76-1737	76-1740	*76-1748	76-1749	*77-1765	77-1777
		77-1809	77-1810	*77-1822	77-1823	*89-2350	89-2360	*89-2403	*90-2428	90-2431
		90-2432	90-2454	*90-2465	90-2466	*93-2532	93-2533	93-2548	93-2550	*93-2558
		*98-2722	98-2725	98-2726	98-2749	*98-2760	98-2761	*109-3175	109-3178	109-3179
		109-3204	*109-3217	109-3218	*116-3536	116-3539	116-3540	*116-3565	*116-3578	116-3579
		*129-4424	129-4425	129-4449	129-4459	*129-4462	129-4463	*130-4482	130-4483	130-4505
		130-4515	*130-4518	*133-4630	133-4631	133-4656	*133-4659	133-4660	*139-5008	139-5011
		139-5012	139-5038	139-5057	*139-5072	139-5073	*140-5090	140-5093	140-5094	140-5117
		*140-5132	140-5133	*166-6866	166-6867	166-6882	166-6885	*166-6895	182-7417	183-7433
		#183-7446								
FPHI	= 000010	#18-685	35-1043							
FMLD	= 000004	#18-684	33-1011	145-5433	146-5490	147-5556	157-6406	158-6458		
GNS	- *****	13-518	13-518	14-553	14-554	14-555	14-556	14-572	89-2416	89-2417
		167-6930	167-6931	167-6932	172-7202	180-7230	180-7230	180-7230	180-7230	180-7230
		180-7230	180-7230	180-7230	180-7230	180-7230	180-7230	180-7230	180-7230	180-7230
		180-7230	180-7230	180-7230	180-7230	180-7230	180-7230	180-7231	180-7231	182-7272

SYMBOL CROSS REFERENCE		REFERENCES									
SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE	SYMBOL	VALUE
		182-7283	182-7285	182-7301	182-7304	182-7356	182-7358	182-7360	182-7362	182-7364	
		182-7366	182-7368	182-7370	182-7372	182-7374	182-7376	182-7378	182-7380	182-7382	
		182-7384	182-7386	182-7388	182-7390						
HIGH1	070000	20-794	#183-7456								
HIT	= 000400	#18-700	154-6282	155-6326	156-6369	157-6420	158-6474	159-6516	160-6559	161-6616	
		162-6679	163-6725	164-6773							
MODE	= 000002	#18-696	80-1938	81-1973	82-2011	83-2053	84-2100	85-2145	86-2192	87-2234	
		88-2279	89-2363	90-2435	91-2475	92-2504	93-2534	94-2573	95-2611	96-2649	
		97-2687	98-2729	99-2773	100-2835	105-2994	106-3041	107-3089	108-3136	109-3180	
		110-3235	111-3293	112-3354	113-3401	114-3449	115-3497	116-3541	117-3596	118-3654	
		119-3708	120-3765	121-3809	122-3866	123-3933	124-3978	125-4073	126-4166	127-4272	
		128-4377	129-4426	130-4484	131-4534	132-4588	133-4633	134-4675	135-4726	136-4797	
		137-4869	138-4940	139-5015	140-5095	141-5141	142-5174	143-5234	144-5334	145-5420	
		146-5476	147-5541	148-5638	149-5762	150-5888	151-5983	152-6076	153-6172	154-6268	
		155-6311	156-6355	157-6395	158-6446	159-6503	160-6545	161-6595	162-6654	163-6708	
		164-6755	165-6798	165-6819	167-6935	167-6942	167-6953	169-7064	170-7096		
HPB	= 004000	#18-703	128-4391	129-4442	130-4498	131-4554	135-4743	136-4815	137-4886	138-4957	
		139-5034	139-5053								
HT	= 000011	173-7216	173-7216								
IBSAVE	046222	#181-7243	*182-7245	182-7259	*182-7266	*182-7270	182-7350				
KPAR0	= 172340	#18-651	*72-1582								
KPAR1	= 172342	#18-652	*72-1583								
KPAR2	= 172344	#18-653	*72-1584								
KPAR3	= 172346	#18-654	*72-1585								
KPAR4	= 172350	#18-655	*76-1712	*77-1762	*79-1879	79-1910	79-1911	*79-1925	*88-2271	88-2301	
		88-2305	*88-2323	*89-2348	89-2382	89-2386	*89-2402	*100-2825	100-2866	*100-2881	
		*162-6647									
KPAR5	= 172352	#18-656									
KPAR6	= 172354	#18-657									
KPAR7	= 172356	#18-658	*72-1586								
KPDR0	= 172300	#18-643	*72-1574								
KPDR1	= 172302	#18-644	*72-1575								
KPDR2	= 172304	#18-645	*72-1576								
KPDR3	= 172306	#18-646	*72-1577								
KPDR4	= 172310	#18-647	*72-1578								
KPDR5	= 172312	#18-648	*72-1579								
KPDR6	= 172314	#18-649	*72-1580								
KPDR7	= 172316	#18-650	*72-1581								
LF	= 000012	173-7216	173-7216								
LOOP	002062	#18-586	*21-813	*21-814	*74-1651	*74-1654	*77-1811	*77-1814	*79-1912	*79-1917	
		*80-1956	*80-1959	*81-1991	*81-1994	*82-2025	*82-2028	*83-2072	*83-2075	*84-2119	
		*84-2122	*85-2164	*85-2167	*86-2211	*86-2214	*87-2251	*87-2254	*88-2306	*88-2311	
		*89-2387	*89-2392	*99-2792	*99-2795	*100-2858	*100-2861	*100-2867	*100-2870	*120-3787	
		*120-3790	*121-3832	*121-3835	*126-4195	*126-4198	*126-4226	*126-4229	*127-4301	*127-4304	
		*127-4332	*127-4335	*148-5681	*148-5684	*148-5713	*148-5716	*149-5806	*149-5809	*149-5838	
		*149-5841	*168-7026	*168-7027							
		20-786	#183-7454								
LOW1	060000	#18-702	128-4399	129-4452	130-4508	131-4562	135-4753	136-4825	137-4896	138-4967	
LFB	= 002000	#72-1574									
MAGPRE	007074	14-547	#18-717	21-805	31-977	32-993	32-997	34-1025	34-1029	38-1091	
OFF	= 001015	74-1627	75-1673	76-170L	77-1761	78-1838	79-1876	80-1936	80-1952	81-1971	
		81-1987	82-2006	82-2024	83-2048	83-2066	84-2095	84-2113	85-2140	85-2158	

SYMBOL CROSS REFERENCE
SYMBOL VALUE

REFERENCES

CREF V01

		86-2187	86-2205	87-2233	87-2247	88-2270	88-2298	89-2339	89-2379	90-2427
		90-2450	91-2473	91-2488	92-2502	92-2517	93-2531	93-2547	94-2566	94-2583
		95-2604	95-2621	96-2642	96-2659	97-2680	97-2697	98-2717	98-2745	99-2772
		99-2787	100-2812	100-2856	100-2890	105-2982	105-3009	106-3053	107-3077	107-3104
		108-3127	108-3148	109-3169	109-3200	110-3226	110-3261	111-3284	111-3319	112-3342
		112-3369	113-3392	113-3413	114-3437	114-3464	115-3488	115-3509	116-3530	116-3561
		117-3587	117-3622	118-3645	118-3680	119-3703	119-3729	120-3763	120-3781	121-3808
		121-3828	122-3859	122-3884	123-3924	123-3945	124-3970	124-4000	124-4024	124-4048
		125-4065	125-4095	125-4121	125-4145	126-4162	126-4190	126-4222	126-4251	127-4268
		127-4296	127-4328	127-4357	128-4375	128-4390	129-4423	129-4441	130-4481	130-4497
		131-4532	131-4548	132-4586	132-4603	133-4629	133-4648	134-4673	134-4690	135-4718
		135-4740	136-4789	136-4812	137-4861	137-4883	138-4932	138-4954	139-5002	139-5033
		140-5089	140-5113	141-5140	142-5169	142-5192	143-5229	143-5261	144-5329	144-5364
		145-5413	145-5442	145-5444	146-5469	146-5497	146-5499	147-5533	147-5571	147-5573
		148-5629	148-5676	148-5708	148-5736	149-5753	149-5801	149-5833	149-5862	150-5879
		150-5919	150-5940	150-5957	151-5974	151-6014	151-6033	151-6050	152-6067	152-6107
		152-6128	152-6146	153-6163	153-6205	153-6226	153-6243	154-6266	154-6281	155-6309
		155-6325	156-6353	156-6368	157-6393	157-6415	158-6444	158-6469	159-6501	159-6515
		160-6543	160-6557	161-6591	161-6615	162-6644	162-6676	163-6707	163-6724	164-6754
		164-6772	165-6787	165-6836	166-6863	166-6881	167-6903	167-6964	168-6991	168-7039
		169-7061	169-7077	170-7093	170-7109	171-7140				
		#18-687	142-5186	143-5253	144-5355	147-5557				
PEA	= 00020C	#18-712								
PEHI	= 000200	#18-711								
PELO	= 000100	182-7297	#182-7392							
PRDATA	050064	182-7275	182-7296	#183-7432						
PRTABL	050420	182-7276	#182-7355							
PRTITL	047056	#18-616	*19-745	*21-804	*165-6797	*168-7031				
PSW	= 177776	176-7222	#180-7230							
RDCHR	= 104407	#180-7230	177-7224	178-7226	#180-7230					
RDDEC	= 104412	#180-7230								
RDLIN	= 104410	*141-5152	141-5156	*142-5198	142-5203	*143-5272	143-5278	*143-5283	143-5289	*143-5294
RDOCT	= 104411	143-5300	*144-5375	144-5381	*144-5386	144-5392	*144-5397	144-5403	*165-6851	165-6856
RECDAT	050520	*166-6884	166-6893	*171-7143	171-7147	182-7419	183-7433	#183-7447		
		#20-794	32-993	80-1936	81-1971	82-2006	83-2048	84-2095	87-2233	88-2270
RELCTH	002452	89-2339	90-2427	91-2473	92-2502	93-2531	94-2566	95-2604	98-2717	99-2772
		100-2812	105-2982	109-3169	110-3226	112-3342	113-3392	116-3530	117-3587	119-3703
		120-3763	122-3859	123-3924	124-3970	126-4162	128-4375	129-4423	130-4481	131-4532
		132-4586	133-4629	134-4673	135-4718	137-4861	139-5002	140-5089	141-5140	142-5169
		143-5229	144-5329	145-5413	146-5469	147-5533	148-5629	150-5879	152-6067	154-6266
		156-6353	157-6393	158-6444	160-6543	161-6591	162-6644	163-6707	164-6754	165-6787
		166-6863	167-6903	168-6991	169-7061	170-7093				
RELCTL	002424	#20-786	34-1025	74-1627	75-1673	76-1708	77-1701	78-1838	79-1876	85-2140
		86-2187	96-2642	97-2680	107-3077	108-3127	111-3284	114-3437	115-3488	118-3645
		121-3808	125-4065	127-4268	136-4789	138-4932	149-5753	151-5974	153-6163	155-6309
		159-6501								
SAVRO	046204	#181-7236	*182-7246	182-7342						
SAVR1	046206	#181-7237	*182-7247	182-7343						
SAVR2	046210	#181-7238	*182-7248	182-7344						
SAVR3	046212	#181-7239	*182-7249	182-7345						
SAVR4	046214	#181-7240	*182-7250	182-7346						

SYMBOL	CROSS REFERENCE	REFERENCES
SYMBOL	VALUE	
SAVRS	= 046216	#181-7241 *182-7251 182-7347
SCOPE	= 000004	#13-539
SCPCND	= 000004	#18-716 23-820 24-842 25-864 26-885 27-906 28-930 29-947 30-960
		31-976 32-993 33-1010 34-1025 35-1042 36-1057 37-1072 38-1087 39-1104
		40-1118 41-1131 42-1145 43-1160 44-1175 45-1190 46-1205 47-1220 48-1235
		49-1250 50-1260 51-1270 52-1280 53-1290 54-1300 55-1310 56-1320 57-1330
		58-1340 59-1350 60-1360 61-1375 62-1392 63-1408 64-1423 65-1438 66-1453
		67-1468 68-1485 69-1504 70-1514 71-1524 72-1540 73-1605 74-1627 75-1673
		76-1708 77-1761 78-1838 79-1876 80-1936 81-1971 82-2006 83-2048 84-2095
		85-2140 86-2187 87-2233 88-2270 89-2339 90-2427 91-2473 92-2502 93-2531
		94-2566 95-2604 96-2642 97-2680 98-2717 99-2772 100-2812 101-2898 102-2919
		103-2937 104-2959 105-2982 106-3032 107-3077 108-3127 109-3169 110-3226 111-3284
		112-3342 113-3392 114-3437 115-3488 116-3530 117-3587 118-3645 119-3703 120-3763
		121-3808 122-3859 123-3924 124-3970 125-4065 126-4162 127-4268 128-4375 129-4423
		130-4481 131-4532 132-4586 133-4629 134-4673 135-4718 136-4789 137-4861 138-4932
		139-5002 140-5089 141-5140 142-5169 143-5229 144-5329 145-5413 146-5469 147-5533
		148-5629 149-5753 150-5879 151-5974 152-6067 153-6163 154-6266 155-6309 156-6353
		157-6393 158-6444 159-6501 160-6543 161-6591 162-6644 163-6707 164-6754 165-6787
		166-6863 167-6903 168-6991 169-7061 170-7093 171-7127 172-7160
SRO	= 177572	#18-659 *76-1717 *76-1732 *77-1780 *77-1800 *79-1890 *79-1900 *88-2282 *88-2296
SR3	= 172516	*89-2353 *89-2377 *100-2839 *100-2854 *100-2887 *162-6652 *162-6674
		#18-660 *76-1718 *76-1733 *77-1781 *77-1801 *79-1891 *79-1901 *88-2283 *88-2297
		*89-2354 *89-2378 *100-2840 *100-2855 *100-2888 *162-6653 *162-6675
START	001000	14-541 #14-543
STRTP	002130	#19-726 *19-769 182-7338 182-7341
STRST	002126	#19-725 19-757 *19-768
SWR	002074	*14-565 #18-593 19-750 19-754 19-761 122-3887 172-7192 182-7252 182-7253
		182-7325 182-7329 182-7348
SWREG	000176	#13-518
TDAR	= 000001	#18-695 80-1938 81-1973 82-2011 83-2053 84-2100 85-2145 86-2192 105-2994
		107-3089 109-3184 109-3189 112-3354 114-3449 116-3545 116-3550 120-3772 126-4166
		127-4272 132-4588 133-4633 134-4675 135-4726 136-4797 137-4869 138-4940 148-5638
		148-5652 149-5762 149-5776 149-5785
		132-4604 133-4649 134-4696 135-4762 136-4834 137-4905 138-4976 140-5114
TPB	= 001000	#18-701
TPE	= 000040	#18-710
TSTCNT	002140	#19-730 *19-763 19-764 *19-766
TSTID	= 000001	#18-715
TSTIMS	002142	#19-731 19-764
TST1	002576	#23-820
TST10	003364	#30-960
TST100	012730	#86-2187
TST101	013154	#87-2233
TST102	013336	#88-2270
TST103	013654	#89-2339
TST104	014432	#90-2427
TST105	014644	#91-2473
TST106	014776	#92-2502
TST107	015134	#93-2531
TST11	003426	#31-976
TST110	015320	#94-2566
TST111	015516	#95-2604
TST112	015716	#96-2642

SYMBOL	CROSS REFERENCE	REFERENCES
SYMBOL	VALUE	
TST113	016114	#97-2680
TST114	016314	#98-2717
TST115	016536	#99-2772
TST116	016732	#100-2812
TST117	017324	#101-2898
TST12	003506	#32-993
TST120	017420	#102-2919
TST121	017476	#103-2937
TST122	017604	#104-2959
TST123	017712	#105-2982
TST124	020130	#106-3032
TST125	020324	#107-3077
TST126	020542	#108-3127
TST127	020744	#109-3169
TST13	003602	#33-1010
TST130	021212	#110-3226
TST131	021470	#111-3284
TST132	021746	#112-3342
TST133	022164	#113-3392
TST134	022366	#114-3437
TST135	022604	#115-3488
TST136	023006	#116-3530
TST137	023254	#117-3587
TST14	003652	#34-1025
TST140	023532	#118-3645
TST141	024010	#119-3703
TST142	024234	#120-3763
TST143	024444	#121-3808
TST144	024644	#122-3859
TST145	025116	#123-3924
TST146	025300	#124-3970
TST147	025620	#125-4065
TST15	003746	#35-1042
TST150	026140	#126-4162
TST151	026534	#127-4268
TST152	027130	#128-4375
TST153	027272	#129-4423
TST154	027464	#130-4481
TST155	027650	#131-4532
TST156	030030	#132-4586
TST157	030156	#133-4629
TST16	004016	#36-1057
TST160	030332	#134-4673
TST161	030476	#135-4718
TST162	030726	#136-4789
TST163	031156	#137-4861
TST164	031412	#138-4932
TST165	031646	#139-5002
TST166	032122	#140-5089
TST167	032326	#141-5140
TST17	004066	#37-1072
TST170	032430	#142-5169

SYMBOL	CROSS REFERENCE VALUE	REFERENCES
TST171	032630	#143-5229
TST172	033166	#144-5329
TST173	033524	#145-5413
TST174	034014	#146-5469
TST175	034304	#147-5533
TST176	034662	#148-5629
TST177	035314	#149-5753
TST2	002672	#24-842
TST20	004136	#38-1087
TST200	035746	#150-5879
TST201	036302	#151-5974
TST202	036636	#152-6067
TST203	037172	#153-6163
TST204	037526	#154-6266
TST205	037650	#155-6309
TST206	037772	#156-6353
TST207	040114	#157-6393
TST21	004216	#39-1104
TST210	040300	#158-6444
TST211	040464	#159-6501
TST212	040606	#160-6543
TST213	040736	#161-6591
TST214	041104	#162-6644
TST215	041316	#163-6707
TST216	041434	#164-6754
TST217	041552	#165-6787
TST22	004262	#40-1118
TST220	042034	#166-6863
TST221	042202	#167-6903
TST222	042732	#168-6991
TST223	043326	#169-7061
TST224	043472	#170-7093
TST225	043640	#171-7127
TST226	043774	#172-7160
TST23	004332	#41-1131
TST24	004402	#42-1145
TST25	004452	#43-1160
TST26	004522	#44-1175
TST27	004572	#45-1190
TST3	002766	#25-864
TST30	004642	#46-1205
TST31	004712	#47-1220
TST32	004762	#48-1235
TST33	005032	#49-1250
TST34	005102	#50-1260
TST35	005152	#51-1270
TST36	005222	#52-1280
TST37	005272	#53-1290
TST4	003062	#26-885
TST40	005342	#54-1300
TST41	005412	#55-1310
TST42	005462	#56-1320

SYMBOL CROSS REFERENCE

SYMBOL	VALUE	REFERENCES
JMPRO6	= 170214	#18-667 *72-1597
JMPRO7	= 170216	#18-668 *72-1598
JMPRO8	= 170220	#18-669 *77-1766 *77-1825
JMPRO9	= 170222	#18-670 *77-1777
VCIP	= 010000	#18-691 101-2902 102-2922 103-2941 103-2944 104-2963 104-2966 105-2986 106-3036
		107-3081 108-3131 109-3173 110-3232 110-3247 110-3250 111-3290 111-3305 111-3308
		112-3346 113-3396 114-3441 115-3492 116-3534 117-3593 117-3608 117-3611 118-3651
		118-3666 118-3669 122-3863 123-3928 131-4552 134-4694 142-5171 143-5231 143-5266
		144-5331 144-5369 145-5415 145-5453 146-5471 146-5509 147-5535 147-5585 150-5883
		151-5978 152-6071 153-6167 157-6418 158-6472 167-6907 168-6995
VLD	= 010000	#18-704 105-3010 106-3054 107-3105 108-3149 109-3201 110-3258 111-3316 112-3370
		113-3414 114-3465 115-3510 116-3562 117-3619 118-3677 122-3890 122-3898 122-3907
		122-3915 123-3948 167-6966
VSIU	= 020000	#18-692 103-2938 103-2947 104-2960 104-2969 105-2983 106-3033 107-3078 108-3128
		109-3170 110-3229 111-3287 112-3343 113-3393 114-3438 115-3489 116-3531 117-3590
		118-3648 122-3860 123-3925 150-5880 151-5975 152-6068 153-6164 167-6904 168-6992
WWWPD	000100	#18-686 131-4538 144-5340 145-5346 145-5427 145-5431 146-5484 146-5488 147-5550
		147-5554 158-6452 158-6458
WWWPT	= 002000	#18-690 134-4681 143-5240 143-5245 157-6401 157-6406
SAPTHD	001452	15-575 #15-575
SASTAT	- *****	17-579 17-579
SATYC	001636	17-579 #17-579
SATY1	001612	#17-579
SATY3	001620	#17-579 173-7216
SATY4	001630	#17-579
SBASE	001542	#16-577
SBELL	002112	#18-602
SBIN	046114	*179-7228 *179-7228 179-7228 #179-7228
SCDW1	001546	#16-577
SCDW2	001550	#16-577
SCHARC	044602	*173-7216 *173-7216 173-7216 #173-7216 #173-7216
SCKSWR	- *****	180-7230
SCNTLG	045533	#176-7222
SCNTLU	045526	#176-7222
SCPUOP	001514	#16-577
SCRLF	002117	#18-604 173-7216 173-7216 173-7216 176-7222 176-7222 178-7226 178-7226
SDBLK	045250	175-7220 #175-7220
SDDW0	001552	#16-577
SDDW1	001554	#16-577
SDDW10	001576	#16-577
SDDW11	001600	#16-577
SDDW12	001602	#16-577
SDDW13	001604	#16-577
SDDW14	001606	#16-577
SDDW15	001610	#16-577
SDDW2	001556	#16-577
SDDW3	001560	#16-577
SDDW4	001562	#16-577
SDDW5	001564	#16-577
SDDW6	001566	#16-577
SDDW7	001570	#16-577
SDDW8	001572	#16-577

SYMBOL	CROSS REFERENCE VALUE	REFERENCES
\$DDW9	001574	#16-577
\$DEVCT	001476	#16-577
\$DEVM	001544	#16-577
\$DCIAGN	044232	172-7197 172-7206 #172-7212
\$DTBL	045240	175-7220 #175-7220
\$ENDAD	044222	13-536 #172-7208
\$ENULL	002122	#18-606 172-7204 182-7287 182-7303
\$ENV	001506	#16-577 17-579 17-579 173-7216 182-7321
\$ENVM	001507	14-563 #16-577 17-579 39-2340 167-6911 168-6999 173-7216 177-7216
\$EROVR	046230	#182-7246 182-7352
\$ERROR	046224	13-529 180-7231 #182-7245
\$ERRPC	002422	*19-740 #19-779
\$ETABL	001506	#16-577
\$ETEND	001612	15-575 #16-577
\$FATAL	001470	#16-577 *17-579 *19-739 *19-744 *182-7265 *182-7271
\$FFLG	002056	*17-579 *17-579 17-579 *17-579 #17-579
\$FILLC	002110	#18-600 173-7216 173-7216 173-7216
\$FILLS	002107	#18-599 173-7216 173-7216
\$GTSWR	*****	180-7230
\$HD	000003	13-517 13-517 13-517
\$HIBTS	001452	#15-575
\$HIOCT	045662	*177-7224 #177-7224
\$ICNT	044240	*172-7195 172-7196 *172-7199 #172-7215
\$LF	002120	#18-605 173-7216 173-7216 176-7222 176-7222 176-7222 178-7226 178-7226
\$LFLG	002055	*17-579 #17-579
\$MADR1	001520	#16-577
\$MADR2	001524	#16-577
\$MADR3	001530	#16-577
\$MADR4	001534	#16-577
\$MAIL	001466	15-575 15-575 #16-577 106-3032 173-7216
\$MAMS1	001516	#16-577
\$MAMS2	001522	#16-577
\$MAMS3	001526	#16-577
\$MAMS4	001532	#16-577
\$MBADR	001454	#15-575
\$MFLG	002054	*17-579 17-579 *17-579 #17-579
\$MNEW	045551	#176-7222
\$MSGAD	001502	#16-577 *17-579 17-579
\$MSGLG	001504	#16-577 *17-579
\$MSGTY	001466	*14-557 #16-577 17-579 *17-579 17-579 *17-579 *182-7327
\$MSWR	045540	#176-7222
\$MTYP1	001517	#16-577
\$MTYP2	001523	#16-577
\$MTYP3	001527	#16-577
\$MTYP4	001533	#16-577
\$NULL	002106	#18-598 173-7216 173-7216 173-7216
\$NWTST	000001	#22-820 22-820 #23-820 23-820 #23-842 23-842 #24-842 24-842 #24-864
		24-864 #25-864 25-864 #25-885 25-885 #26-885 26-885 #26-906 26-906 #26-926
		#27-906 27-906 #27-930 27-930 #28-930 28-930 #28-947 28-947 #29-947
		29-947 #29-960 29-960 #30-960 30-960 #30-976 30-976 #31-976 31-976 #31-976
		#31-993 31-993 #32-993 32-993 #32-1010 32-1010 #33-1010 33-1010 #33-1025
		33-1025 #34-1025 34-1025 #34-1042 34-1042 #35-1042 35-1042 #35-1057 35-1057

SYMBOL CROSS REFERENCE
 SYMBOL VALUE

REFERENCES

REFERENCES	CREF	VO1
#36-1057	36-1057	#36-1072
38-1087	#38-1104	36-1072
#40-1131	40-1131	#39-1104
42-1160	#43-1160	39-1104
#45-1190	45-1190	41-1145
47-1220	#47-1235	#43-1175
#49-1260	49-1260	43-1175
#52-1290	52-1290	#46-1205
#55-1320	55-1320	46-1205
#58-1350	58-1350	#48-1250
61-1375	#61-1392	48-1250
#63-1423	63-1423	#50-1260
65-1453	#66-1453	#50-1270
#68-1485	68-1485	50-1270
#70-1524	70-1524	#53-1300
#73-1605	73-1605	53-1300
75-1673	#75-1708	#56-1320
#77-1838	77-1838	#59-1350
79-1936	#80-1936	#59-1360
#82-2006	82-2006	#62-1392
84-2095	#84-2140	62-1392
#86-2233	86-2233	#64-1423
88-2339	#89-2339	64-1423
#91-2473	91-2473	#66-1468
93-2531	#93-2566	66-1468
#95-2642	95-2642	#68-1504
97-2717	#98-2717	68-1504
#100-2812	100-2812	#71-1524
102-2919	#102-2937	#71-1540
#104-2982	104-2982	#73-1627
106-3077	#107-3077	73-1627
#109-3169	109-3169	#76-1708
111-3284	#111-3342	76-1708
#113-3437	113-3437	#78-1838
115-3530	#116-3530	78-1838
#118-3645	118-3645	#80-1971
120-3763	#120-3808	80-1971
#122-3924	122-3924	#82-2048
#125-4065	125-4065	82-2048
127-4268	#127-4375	#85-2140
#129-4481	129-4481	85-2140
131-4586	#132-4586	#87-2233
#134-4673	134-4673	87-2233
136-4789	#136-4861	#87-2270
#138-5002	138-5002	87-2270
140-5140	#141-5140	#89-2427
#143-5229	143-5229	89-2427
145-5413	#145-5469	#92-2502
#147-5629	147-5629	92-2502
149-5879	#150-5879	#94-2566
#152-6067	152-6067	94-2566
154-6266	#154-6309	#96-2680
		96-2680
		#98-2772
		98-2772
		#99-2772
		99-2772
		#101-2898
		101-2898
		#103-2937
		103-2937
		#105-2982
		105-2982
		#105-3032
		105-3032
		#106-3032
		106-3032
		#108-3127
		108-3127
		#110-3226
		110-3226
		#110-3284
		110-3284
		#112-3342
		112-3342
		#112-3392
		112-3392
		#113-3392
		113-3392
		#115-3488
		115-3488
		#117-3587
		117-3587
		#119-3703
		119-3703
		#119-3763
		119-3763
		#122-3859
		122-3859
		#124-4065
		124-4065
		#126-4268
		126-4268
		#128-4423
		128-4423
		#131-4532
		131-4532
		#133-4629
		133-4629
		#135-4718
		135-4718
		#137-4861
		137-4861
		#139-5089
		139-5089
		#142-5169
		142-5169
		#144-5329
		144-5329
		#146-5469
		146-5469
		#148-5753
		148-5753
		#151-5974
		151-5974
		#153-6163
		153-6163
		#155-6353
		155-6353
		#37-1072
		37-1072
		#37-1087
		37-1087
		#39-1118
		39-1118
		#42-1145
		42-1145
		#44-1175
		44-1175
		#46-1220
		46-1220
		#49-1250
		49-1250
		#51-1280
		51-1280
		#54-1310
		54-1310
		#57-1340
		57-1340
		#60-1375
		60-1375
		#62-1408
		62-1408
		#65-1438
		65-1438
		#67-1468
		67-1468
		#69-1514
		69-1514
		#72-1540
		72-1540
		#74-1673
		74-1673
		#76-1761
		76-1761
		#79-1876
		79-1876
		#81-1971
		81-1971
		#83-2095
		83-2095
		#86-2187
		86-2187
		#88-2270
		88-2270
		#90-2473
		90-2473
		#92-2531
		92-2531
		#95-2604
		95-2604
		#97-2680
		97-2680
		#99-2812
		99-2812
		#101-2919
		101-2919
		#104-2959
		104-2959
		#106-3032
		106-3032
		#108-3169
		108-3169
		#110-3284
		110-3284
		#113-3392
		113-3392
		#115-3530
		115-3530
		#117-3645
		117-3645
		#120-3763
		120-3763
		#122-3859
		122-3859
		#124-4065
		124-4065
		#127-4268
		127-4268
		#129-4423
		129-4423
		#131-4586
		131-4586
		#133-4673
		133-4673
		#136-4789
		136-4789
		#138-4932
		138-4932
		#140-5140
		140-5140
		#142-5229
		142-5229
		#145-5413
		145-5413
		#147-5533
		147-5533
		#149-5753
		149-5753
		#151-6067
		151-6067
		#153-6266
		153-6266
		#156-6353
		156-6353

SYMBOL	CROSS REFERENCE VALUE	REFERENCES
		#156-6393 156-6393 #157-6393 157-6393 #157-6444 157-6444 #158-6444 158-6444 #158-6501 158-6501 #159-6501 159-6501 #159-6543 159-6543 #160-6543 160-6543 #160-6591 160-6591 #161-6591 161-6591 #161-6644 161-6644 #162-6644 162-6644 #162-6707 162-6707 #163-6707 163-6707 #163-6754 163-6754 #164-6754 164-6754 #164-6787 164-6787 #165-6787 165-6787 #165-6863 165-6863 #166-6863 166-6863 #166-6903 166-6903 #167-6903 167-6903 #167-6991 167-6991 #168-6991 168-6991 #168-7061 168-7061 #169-7061 169-7061 #169-7093 169-7093 #170-7093 170-7093 #170-7127 170-7127 #171-7127 171-7127 #171-7160 171-7160 #172-7160 172-7160
\$OCNT	045030	*174-7218 #174-7218 #174-7218 174-7218 *174-7218 *174-7218 #174-7218
\$OMODE	045032	*174-7218 *174-7218 #16-577 19-759 89-2412 167-6926 172-7190 *172-720U *172-7201 172-7203
\$PASS	001474	*14-562 #15-575
\$PASTM	001460	#15-575
\$QUES	002116	#18-603 173-7216 173-7216 176-7222 176-7222 176-7222 178-7226 178-7226 178-7226
\$RDCHR	045260	#176-7222 180-7230 180-7230
\$RDDEC	045664	#178-7226 180-7230 180-7230
\$RDLIN	045410	#176-7222 180-7230 180-7230
\$RDOCT	045562	#177-7224 180-7230 180-7230
\$RDSZ	= 000010	#176-7222 176-7222
\$RZA	*****	180-7230
\$SAVRE	*****	180-7230
\$SCPSE	002144	13-526 #19-735
\$SETUP	- 000000	#13-437 176-7222 176-7222
\$SWR	- 160000	13-517 #13-517 23-820 24-842 25-864 26-885 27-906 28-930 29-947
		30-960 31-976 32-993 33-1010 34-1025 35-1042 36-1057 37-1072 38-1087
		39-1104 40-1118 41-1131 42-1145 43-1160 44-1175 45-1190 46-1205 47-1220
		48-1235 49-1250 50-1260 51-1270 52-1280 53-1290 54-1300 55-1310 56-1320
		57-1330 58-1340 59-1350 60-1360 61-1375 62-1392 63-1408 64-1423 65-1438
		66-1453 67-1468 68-1485 69-1504 70-1514 71-1524 72-1540 73-1605 74-1627
		75-1673 76-1708 77-1761 78-1838 79-1876 80-1936 81-1971 82-2006 83-2048
		84-2095 85-2140 86-2187 87-2233 88-2270 89-2339 90-2427 91-2473 92-2502
		93-2531 94-2566 95-2604 96-2642 97-2680 98-2717 99-2772 100-2812 101-2898
		102-2919 103-2937 104-2959 105-2982 106-3032 107-3077 108-3127 109-3169 110-3226
		111-3284 112-3342 113-3392 114-3437 115-3488 116-3530 117-3587 118-3645 119-3703
		120-3763 121-3808 122-3859 123-3924 124-3970 125-4065 126-4162 127-4268 128-4375
		129-4423 130-4481 131-4532 132-4586 133-4629 134-4673 135-4718 136-4789 137-4861
		138-4932 139-5002 140-5089 141-5140 142-5169 143-5229 144-5329 145-5413 146-5469
		147-5533 148-5629 149-5753 150-5879 151-5974 152-6067 153-6163 154-6266 155-6309
		156-6353 157-6393 158-6444 159-6501 160-6543 161-6591 162-6644 163-6707 164-6754
		165-6787 166-6863 167-6903 168-6991 169-7061 170-7093 171-7127 172-7160
\$SWREG	001510	14-565 #16-577
\$TESTN	001472	#16-577 19-746 19-747 19-752 *19-756 *21-906 *23-837 *24-859 *25-880
		*26-901 *27-922 *28-940 *29-955 *30-970 *31-987 *32-1004 *33-1019 *34-1036
		*35-1051 *36-1066 *37-1081 *38-1098 *39-1112 *40-1127 *41-1140 *42-1155 *43-1170
		*44-1185 *45-1200 *46-1215 *47-1230 *48-1245 *49-1259 *50-1269 *51-1279 *52-1289
		*53-1299 *54-1309 *55-1319 *56-1329 *57-1339 *58-1349 *59-1359 *60-1369 *61-1387
		*62-1404 *63-1419 *64-1434 *65-1449 *66-1464 *67-1481 *68-1499 *69-1513 *70-1523
		*71-1533 *72-1568 *73-1620 *74-1666 *75-1701 *76-1754 *77-1831 *78-1870 *79-1932
		*80-1967 *81-2002 *82-2044 *83-2090 *84-2135 *85-2182 *86-2227 *87-2262 *88-2328
		*89-2418 *90-2468 *91-2497 *92-2526 *93-2561 *94-2599 *95-2637 *96-2675 *97-2713
		*98-2764 *99-2804 *100-2894 *101-2915 *102-2932 *103-2954 *104-2976 *105-3026 *106-3032
		*106-3071 *107-3121 *108-3165 *109-3220 *110-3278 *111-3336 *112-3386 *113-3431 *114-3482
		*115-3526 *116-3581 *117-3639 *118-3697 *119-3755 *120-3803 *121-3845 *122-3923 *123-3955

SYMBOL CROSS REFERENCE
SYMBOL VALUE

\$TIMES 044236
\$IKB 002100
\$TKS 002076
\$TA = 000227

REFERENCES

*124-4050	*125-4147	*126-4253	*127-4359	*128-4407	*129-4465	*130-4520	*131-4570	*132-4612
*133-4662	*134-4704	*135-4775	*136-4847	*137-4918	*138-4989	*139-5076	*140-5135	*141-5158
*142-5205	*143-5302	*144-5405	*145-5461	*146-5518	*147-5614	*148-5738	*149-5864	*150-5959
*151-6052	*152-6148	*153-6245	*154-6289	*155-6333	*156-6376	*157-6427	*158-6481	*159-6523
*160-6566	*161-6623	*162-6686	*163-6732	*164-6780	*165-6858	*166-6898	*167-6986	*168-7056
*169-7088	*170-7120	*171-7154	*172-7188	*172-7198	182-7284			
172-7196	#172-7214							
#18-595	173-7216	173-7216	173-7216	173-7216	176-7222	176-7222	176-7222	176-7222
#18-594	173-7216	173-7216	173-7216	173-7216	176-7222	176-7222	176-7222	176-7222
13-517	#13-517	22-820	23-820	#23-820	23-842	24-842	#24-842	24-864
25-864	#25-864	25-885	26-885	#26-885	26-906	27-906	#27-906	27-930
28-930	#28-930	28-947	29-947	#29-947	29-960	30-960	#30-960	30-976
31-976	#31-976	31-993	32-993	#32-993	32-1010	33-1010	#33-1010	33-1025
34-1025	#34-1025	34-1042	35-1042	#35-1042	35-1057	36-1057	#36-1057	36-1072
37-1072	#37-1072	37-1087	38-1087	#38-1087	38-1104	39-1104	#39-1104	39-1118
40-1118	#40-1118	40-1131	41-1131	#41-1131	41-1145	42-1145	#42-1145	42-1160
43-1160	#43-1160	43-1175	44-1175	#44-1175	44-1190	45-1190	#45-1190	45-1205
46-1205	#46-1205	46-1220	47-1220	#47-1220	47-1235	48-1235	#48-1235	48-1250
49-1250	#49-1250	49-1260	50-1260	#50-1260	50-1270	51-1270	#51-1270	51-1280
52-1280	#52-1280	52-1290	53-1290	#53-1290	53-1300	54-1300	#54-1300	54-1310
55-1310	#55-1310	55-1320	56-1320	#56-1320	56-1330	57-1330	#57-1330	57-1340
58-1340	#58-1340	58-1350	59-1350	#59-1350	59-1360	60-1360	#60-1360	60-1375
61-1375	#61-1375	61-1392	62-1392	#62-1392	62-1408	63-1408	#63-1408	63-1423
64-1423	#64-1423	64-1438	65-1438	#65-1438	65-1453	66-1453	#66-1453	66-1468
67-1468	#67-1468	67-1485	68-1485	#68-1485	68-1504	69-1504	#69-1504	69-1514
70-1514	#70-1514	70-1524	71-1524	#71-1524	71-1540	72-1540	#72-1540	72-1605
73-1605	#73-1605	73-1627	74-1627	#74-1627	74-1673	75-1673	#75-1673	75-1708
76-1708	#76-1708	76-1761	77-1761	#77-1761	77-1838	78-1838	#78-1838	78-1876
79-1876	#79-1876	79-1936	80-1936	#80-1936	80-1971	81-1971	#81-1971	81-2006
82-2006	#82-2006	82-2048	83-2048	#83-2048	83-2095	84-2095	#84-2095	84-2140
85-2140	#85-2140	85-2187	86-2187	#86-2187	86-2233	87-2233	#87-2233	87-2270
88-2270	#88-2270	88-2339	89-2339	#89-2339	89-2427	90-2427	#90-2427	90-2473
91-2473	#91-2473	91-2502	92-2502	#92-2502	92-2531	93-2531	#93-2531	93-2566
94-2566	#94-2566	94-2604	95-2604	#95-2604	95-2642	96-2642	#96-2642	96-2680
97-2680	#97-2680	97-2717	98-2717	#98-2717	98-2772	99-2772	#99-2772	99-2812
100-2812	#100-2812	100-2898	101-2898	#101-2898	101-2919	102-2919	#102-2919	102-2937
103-2937	#103-2937	103-2959	104-2959	#104-2959	104-2982	105-2982	#105-2982	105-3032
106-3032	#106-3032	106-3032	106-3077	107-3077	#107-3077	107-3127	108-3127	#108-3127
108-3169	109-3169	#109-3169	109-3226	110-3226	#110-3226	110-3284	111-3284	#111-3284
111-3342	112-3342	#112-3342	112-3392	113-3392	#113-3392	113-3437	114-3437	#114-3437
114-3488	115-3488	#115-3488	115-3530	116-3530	#116-3530	116-3587	117-3587	#117-3587
117-3645	118-3645	#118-3645	118-3703	119-3703	#119-3703	119-3763	120-3763	#120-3763
120-3808	121-3808	#121-3808	121-3859	122-3859	#122-3859	122-3924	123-3924	#123-3924
123-3970	124-3970	#124-3970	124-4065	125-4065	#125-4065	125-4162	126-4162	#126-4162
126-4268	127-4268	#127-4268	127-4375	128-4375	#128-4375	128-4423	129-4423	#129-4423
129-4481	130-4481	#130-4481	130-4532	131-4532	#131-4532	131-4586	132-4586	#132-4586
132-4629	133-4629	#133-4629	133-4673	134-4673	#134-4673	134-4718	135-4718	#135-4718
135-4789	136-4789	#136-4789	136-4861	137-4861	#137-4861	137-4932	138-4932	#138-4932
138-5002	139-5002	#139-5002	139-5089	140-5089	#140-5089	140-5140	141-5140	#141-5140
141-5169	142-5169	#142-5169	142-5229	143-5229	#143-5229	143-5329	144-5329	#144-5329
144-5413	145-5413	#145-5413	145-5469	146-5469	#146-5469	146-5533	147-5533	#147-5533
147-5629	148-5629	#148-5629	148-5753	149-5753	#149-5753	149-5879	150-5879	#150-5879

SYMBOL CROSS REFERENCE
 SYMBOL VALUE

REFERENCES

CRFF V01

		150-5974	151-5974	#151-5974	151-6067	152-6067	#152-6067	152-6163	153-6163	#153-6163
		153-6266	154-6266	#154-6266	154-6309	155-6309	#155-6309	155-6353	156-6353	#156-6353
		156-6393	157-6393	#157-6393	157-6444	158-6444	#158-6444	158-6501	159-6501	#159-6501
		159-6543	160-6543	#160-6543	160-6591	161-6591	#161-6591	161-6644	162-6644	#162-6644
		162-6707	163-6707	#163-6707	163-6754	164-6754	#164-6754	164-6787	165-6787	#165-6787
		165-6863	166-6863	#166-6863	166-6903	167-6903	#167-6903	167-6991	168-6991	#168-6991
		168-7061	169-7061	#169-7061	169-7093	170-7093	#170-7093	170-7127	171-7127	#171-7127
		171-7160	172-7160	#172-7160						
\$TPB	002104	#18-597	173-7216		173-7216					
\$TPFLG	002111	#18-601	173-7216		173-7216					
\$TPS	002102	#18-596	173-7216		173-7216					
\$TRAP	046116	13-531	#180-7230							
\$TRAP2	046140	#180-7230	180-7230							
\$TRP	= 000014	#180-7230	180-7230	180-7230	180-7230	180-7230	#180-7230	180-7230	180-7230	180-7230
		180-7230	#180-7230	180-7230	180-7230	180-7230	180-7230	#180-7230	180-7230	180-7230
		180-7230	180-7230	#180-7230	180-7230	180-7230	180-7230	180-7230	#180-7230	180-7230
		180-7230	180-7230	180-7230	#180-7230	180-7230	180-7230	180-7230	180-7230	#180-7230
		180-7230	180-7230	180-7230	180-7230	180-7230	#180-7230	180-7230	180-7230	180-7230
		#180-7230	180-7230	180-7230	180-7230	#180-7230	#180-7230	180-7230	180-7230	180-7230
		180-7231	#180-7231							
		180-7230	#180-7230							
\$TRPAD	046152									
\$STSM	001456	#15-575								
\$STNM	002060	#18-585	*19-746	182-7252						
\$STVIN	045516	176-7222	176-7222	176-7222	#176-7222					
\$STYPBN	046042	#179-7228	180-7230	180-7230						
\$STYPDS	045034	#175-7220	180-7230	180-7230						
\$STYPE	044242	17-579	#173-7216	180-7230	180-7230					
\$STYPEC	044454	173-7216	173-7216	173-7216	#173-7216					
\$STYPEX	044604	173-7216	173-7216	173-7216	#173-7216					
\$STYPOC	044632	#174-7218	180-7230	180-7230						
\$STYPON	044646	174-7218	#174-7218	180-7230						
\$STYPOS	044606	*174-7218	180-7230							
\$SUNIT	001500	#16-577								
\$SUNITM	001462	#15-575								
\$SUSWR	001512	#16-577	89-2342	167-6913	168-7001					
\$VECT1	001536	#16-577								
\$VECT2	001540	#16-577								
\$XOFF	= 000023	173-7216	173-7216							
\$XON	= 000021	173-7216	173-7216	173-7216	176-7222					
\$OFILL	045031	*174-7218	*174-7218	174-7218	#174-7218					
.\$ASTA	*****	17-579	17-579							
.\$X	= 001452	#15-575	15-575							

MACRO CROSS REFERENCE

MACRO NAME REFERENCES
 ENDTST

REF V01

#13-512	23-837	24-859	25-880	26-901	27-922	28-940	29-955	30-970	31-987
32-1004	33-1019	34-1036	35-1051	36-1066	37-1081	38-1098	39-1112	40-1127	41-1140
42-1155	43-1170	44-1185	45-1200	46-1215	47-1230	48-1245	49-1259	50-1269	51-1279
52-1289	53-1299	54-1309	55-1319	56-1329	57-1339	58-1349	59-1359	60-1369	61-1387
62-1404	63-1419	64-1434	65-1449	66-1464	67-1481	68-1499	69-1513	70-1523	71-1533
72-1568	73-1620	74-1666	75-1701	76-1754	77-1831	78-1870	79-1932	80-1967	81-2002
82-2044	83-2090	84-2135	85-2182	86-2227	87-2262	88-2328	89-2418	90-2468	91-2497
92-2526	93-2561	94-2599	95-2637	96-2675	97-2713	98-2764	99-2804	100-2894	101-2915
102-2932	103-2954	104-2976	105-3026	106-3071	107-3121	108-3165	109-3220	110-3278	111-3336
112-3386	113-3431	114-3482	115-3526	116-3581	117-3639	118-3697	119-3755	120-3803	121-3845
122-3923	123-3955	124-4050	125-4147	126-4253	127-4359	128-4407	129-4465	130-4520	131-4570
132-4612	133-4662	134-4704	135-4775	136-4847	137-4918	138-4989	139-5076	140-5135	141-5158
142-5205	143-5302	144-5405	145-5461	146-5518	147-5614	148-5738	149-5864	150-5959	151-6052
152-6148	153-6245	154-6289	155-6333	156-6376	157-6427	158-6481	159-6523	160-6566	161-6623
162-6686	163-6732	164-6780	165-6858	166-6898	167-6986	168-7056	169-7088	170-7120	171-7154
172-7188									

ERR #13-442	23-831	24-853	25-875	26-896	27-917	28-935	29-951	30-964	31-983
32-1000	33-1015	34-1032	35-1047	36-1062	37-1077	38-1094	40-1123	41-1136	42-1150
43-1165	44-1180	45-1195	46-1210	47-1225	48-1240	49-1255	50-1265	51-1275	52-1285
53-1295	54-1305	55-1315	56-1325	57-1335	58-1345	59-1355	60-1365	61-1381	62-1398
63-1415	64-1430	65-1445	66-1460	67-1477	68-1495	69-1509	70-1519	71-1529	72-1562
73-1614	74-1656	75-1690	76-1742	77-1816	78-1859	79-1919	80-1961	81-1996	82-2032
83-2077	84-2124	85-2169	86-2216	87-2256	88-2313	89-2394	90-2456	91-2491	92-2520
93-2551	94-2588	95-2626	96-2664	97-2702	98-2751	99-2797	100-2873	101-2910	102-2927
103-2950	104-2972	105-3015	106-3060	107-3110	108-3154	109-3206	110-3265	111-3323	112-3375
113-3420	114-3471	115-3515	116-3567	117-3626	118-3684	119-3735	119-3747	120-3793	121-3838
122-3892	122-3900	122-3909	122-3917	123-3950	124-4006	124-4031	125-4102	125-4128	126-4202
126-4233	127-4308	127-4339	128-4393	128-4401	129-4444	129-4454	130-4500	130-4510	131-4556
131-4564	132-4606	133-4651	134-4698	135-4745	135-4755	135-4764	136-4817	136-4827	136-4836
137-4888	137-4898	137-4907	138-4959	138-4969	138-4978	139-5040	139-5059	140-5119	141-5153
142-5199	143-5273	143-5284	143-5295	144-5376	144-5387	144-5398	145-5457	146-5513	147-5589
147-5596	147-5603	147-5610	148-5688	148-5720	149-5813	149-5845	150-5923	150-5944	151-6018
151-6037	152-6111	152-6132	153-6209	153-6230	154-6284	155-6328	156-6371	157-6422	158-6476
159-6518	160-6561	161-6618	162-6681	163-6727	164-6775	165-6842	165-6852	166-6886	167-6971
168-7042	168-7051	169-7083	170-7115	171-7144	172-7182				

LNOP #13-438	#23-826	#24-848	#25-870	#26-891	#27-912	#28-932	#29-949	#30-962	#31-980
#32-995	#33-1012	#34-1028	#35-1044	#36-1059	#37-1074	#38-1090	#39-1106	#40-1120	#41-1133
#42-1147	#43-1162	#44-1177	#45-1192	#46-1207	#47-1222	#48-1237	#49-1252	#50-1262	#51-1272
#52-1282	#53-1292	#54-1302	#55-1312	#56-1322	#57-1332	#58-1342	#59-1352	#60-1362	#61-1378
#62-1395	#63-1411	#64-1426	#65-1441	#66-1456	#67-1473	#68-1491	#69-1506	#70-1516	#71-1526
#72-1559	#73-1611	#74-1643	#75-1682	#76-1734	#77-1802	#78-1851	#79-1902	#80-1950	#81-1985
#82-2022	#83-2064	#84-2111	#85-2156	#86-2203	#87-2245	#88-2295	#89-2376	#90-2448	#91-2486
#92-2515	#93-2545	#94-2581	#95-2619	#96-2657	#97-2695	#98-2743	#99-2785	#100-2853	#101-2907
#102-2926	#103-2946	#104-2968	#105-3007	#106-3051	#107-3102	#108-3146	#109-3198	#110-3252	#111-3310
#112-3367	#113-3411	#114-3462	#115-3507	#116-3559	#117-3613	#118-3671	#119-3727	#120-3782	#121-3826
#122-3885	#123-3946	#124-3997	#125-4092	#126-4186	#127-4292	#128-4388	#129-4439	#130-4495	#131-4547
#132-4601	#133-4646	#134-4689	#135-4738	#136-4810	#137-4881	#138-4952	#139-5031	#140-5111	#141-5148
#142-5191	#143-5260	#144-5363	#145-5446	#146-5502	#147-5578	#148-5672	#149-5797	#150-5915	#151-6010
#152-6103	#153-6201	#154-6280	#155-6324	#156-6367	#157-6416	#158-6470	#159-6514	#160-6558	#161-6614
#162-6673	#163-6723	#164-6771	#165-6826	#166-6880	#167-6963	#168-7038	#169-7076	#170-7108	#171-7153
#172-7177									

MSG #22-816	23-820	#23-838	24-842	#24-860	25-864	#25-881	26-885	#26-902	27-906
-------------	--------	---------	--------	---------	--------	---------	--------	---------	--------

MACRO CROSS REFERENCE
 MACRO NAME REFERENCES

CREF V01

	#27-923	28-930	#28-941	29-947	#29-956	30-960	#30-971	31-976	#31-988	32-993
	#32-1005	33-1010	#33-1020	34-1025	#34-1037	35-1042	#35-1052	36-1057	#36-1067	37-1072
	#37-1082	38-1087	#38-1099	39-1104	#39-1113	40-1118	#40-1128	41-1131	#41-1141	42-1145
	#42-1156	43-1160	#43-1171	44-1175	#44-1186	45-1190	#45-1201	46-1205	#46-1216	47-1220
	#47-1231	48-1235	#48-1246	49-1250	#60-1370	61-1375	#61-1388	62-1392	#62-1405	63-1408
	#63-1420	64-1423	#64-1435	65-1438	#65-1450	66-1453	#66-1465	67-1468	#67-1482	68-1485
	#68-1500	69-1504	#71-1534	72-1540	#72-1599	73-1605	#73-1621	74-1627	#74-1667	75-1673
	#75-1702	76-1708	#76-1755	77-1761	#77-1832	78-1838	#78-1871	79-1876	#79-1933	80-1936
	#80-1968	81-1971	#81-2003	82-2006	#82-2045	83-2048	#83-2091	84-2095	#84-2136	85-2140
	#85-2183	86-2187	#86-2228	87-2233	#87-2263	88-2270	#88-2329	89-2339	#89-2419	90-2427
	#90-2469	91-2473	#91-2498	92-2502	#92-2527	93-2531	#93-2562	94-2566	#94-2600	95-2604
	#95-2638	96-2642	#96-2676	97-2680	#97-2714	98-2717	#98-2765	99-2772	#99-2805	100-2812
	#100-2895	101-2898	#101-2916	102-2919	#102-2933	103-2937	#103-2955	104-2959	#104-2977	105-2982
	#105-3027	106-3032	#106-3072	107-3077	#107-3122	108-3127	#108-3166	109-3169	#109-3221	110-3226
	#110-3279	111-3284	#111-3337	112-3342	#112-3387	113-3392	#113-3432	114-3437	#114-3483	115-3488
	#115-3527	116-3530	#116-3582	117-3587	#117-3640	118-3645	#118-3698	119-3703	#119-3756	120-3763
	#120-3804	121-3808	#121-3846	122-3859	#123-3956	124-3970	#124-4051	125-4065	#125-4148	126-4162
	#126-4254	127-4268	#127-4360	128-4375	#128-4408	129-4423	#129-4466	130-4481	#130-4521	131-4532
	#131-4571	132-4586	#132-4613	133-4629	#133-4663	134-4673	#134-4705	135-4718	#135-4776	136-4789
	#136-4848	137-4861	#137-4919	138-4932	#138-4990	139-5002	#139-5077	140-5089	#140-5136	141-5140
	#141-5159	142-5169	#142-5206	143-5229	#143-5303	144-5329	#144-5406	145-5413	#145-5462	146-5469
	#146-5519	147-5533	#147-5615	148-5629	#148-5739	149-5753	#149-5865	150-5879	#150-5960	151-5974
	#151-6053	152-6067	#152-6149	153-6163	#153-6246	154-6266	#154-6290	155-6309	#155-6334	156-6353
	#156-6377	157-6393	#157-6428	158-6444	#158-6482	159-6501	#159-6524	160-6543	#160-6567	161-6591
	#161-6624	162-6644	#162-6687	163-6707	#163-6733	164-6754	#164-6781	165-6787	#165-6859	166-6863
	#166-6899	167-6903	#167-6987	168-6991	#168-7057	169-7061	#169-7089	170-7093	#170-7121	171-7127
	#171-7155	172-7160								
NEWTST	#13-433	22-820	23-842	24-864	25-885	26-906	27-930	28-947	29-960	30-976
	31-993	32-1010	33-1025	34-1042	35-1057	36-1072	37-1087	38-1104	39-1118	40-1131
	41-1145	42-1160	43-1175	44-1190	45-1205	46-1220	47-1235	48-1250	49-1260	50-1270
	51-1280	52-1290	53-1300	54-1310	55-1320	56-1330	57-1340	58-1350	59-1360	60-1375
	61-1392	62-1408	63-1423	64-1438	65-1453	66-1468	67-1485	68-1504	69-1514	70-1524
	71-1540	72-1605	73-1627	74-1673	75-1708	76-1761	77-1838	78-1876	79-1936	80-1971
	81-2006	82-2048	83-2095	84-2140	85-2187	86-2233	87-2270	88-2339	89-2427	90-2473
	91-2502	92-2531	93-2566	94-2604	95-2642	96-2680	97-2717	98-2772	99-2812	100-2898
	101-2919	102-2937	103-2959	104-2982	105-3032	106-3077	107-3127	108-3169	109-3226	110-3284
	111-3342	112-3392	113-3437	114-3488	115-3530	116-3587	117-3645	118-3703	119-3763	120-3808
	121-3859	122-3924	123-3970	124-4065	125-4162	126-4268	127-4375	128-4423	129-4481	130-4532
	131-4586	132-4629	133-4673	134-4718	135-4789	136-4861	137-4932	138-5002	139-5089	140-5140
	141-5169	142-5229	143-5329	144-5413	145-5469	146-5533	147-5629	148-5753	149-5879	150-5974
	151-6067	152-6163	153-6266	154-6309	155-6353	156-6393	157-6444	158-6501	159-6543	160-6591
	161-6644	162-6707	163-6754	164-6787	165-6863	166-6903	167-6991	168-7061	169-7093	170-7127
	171-7160									
POP	#13-435	17-579	17-579	175-7220	177-7224	178-7226				
PUSH	#13-435	17-579	17-579	17-579	175-7220	177-7224	178-7226			
SCPSET	#13-447	23-820	24-842	25-864	26-885	27-906	28-930	29-947	30-960	31-976
	32-993	33-1010	34-1025	35-1042	36-1057	37-1072	38-1087	39-1104	40-1118	41-1131
	42-1145	43-1160	44-1175	45-1190	46-1205	47-1220	48-1235	49-1250	50-1260	51-1270
	52-1280	53-1290	54-1300	55-1310	56-1320	57-1330	58-1340	59-1350	60-1360	61-1375
	62-1392	63-1408	64-1423	65-1438	66-1453	67-1468	68-1485	69-1504	70-1514	71-1524
	72-1540	73-1605	74-1627	75-1673	76-1708	77-1761	78-1838	79-1876	80-1936	81-1971
	82-2006	83-2048	84-2095	85-2140	86-2187	87-2233	88-2270	89-2339	90-2427	91-2473

MACRO CROSS REFERENCE
MACRO NAME REFERENCES

	92-2502	93-2531	94-2566	95-2604	96-2642	97-2680	98-2717	99-2772	100-2812	101-2898
	102-2919	103-2937	104-2959	105-2982	106-3032	107-3077	108-3127	109-3169	110-3226	111-3284
	112-3342	113-3392	114-3437	115-3488	116-3530	117-3587	118-3645	119-3703	120-3763	121-3808
	122-3859	123-3924	124-3970	125-4065	126-4162	127-4268	128-4375	129-4423	130-4481	131-4532
	132-4586	133-4629	134-4673	135-4718	136-4789	137-4861	138-4932	139-5002	140-5089	141-5140
	142-5169	143-5229	144-5329	145-5413	146-5469	147-5533	148-5629	149-5753	150-5879	151-5974
	152-6067	153-6163	154-6266	155-6309	156-6353	157-6393	158-6444	159-6501	160-6543	161-6591
	162-6644	163-6707	164-6754	165-6787	166-6863	167-6903	168-6991	169-7061	170-7093	171-7127
	172-7160									
SETTRA	#180-7230	#180-7230	#180-7230	#180-7230	#180-7230	#180-7230	#180-7230	#180-7230	#180-7230	#180-7230
	#180-7230	#180-7231								
STARS	#13-433	#13-434	15-575	15-575	15-575	16-577	17-579	18-581	18-583	19-718
	19-724	20-781	20-783	23-820	23-820	24-842	24-842	25-864	25-864	26-885
	26-885	27-906	27-906	28-930	28-930	29-947	29-947	30-960	30-960	31-976
	31-976	32-993	32-993	33-1010	33-1010	34-1025	34-1025	35-1042	35-1042	36-1057
	36-1057	37-1072	37-1072	38-1087	38-1087	39-1104	39-1104	40-1118	40-1118	41-1131
	41-1131	42-1145	42-1145	43-1160	43-1160	44-1175	44-1175	45-1190	45-1190	46-1205
	46-1205	47-1220	47-1220	48-1235	48-1235	49-1250	49-1250	50-1260	50-1260	51-1270
	51-1270	52-1280	52-1280	53-1290	53-1290	54-1300	54-1300	55-1310	55-1310	56-1320
	56-1320	57-1330	57-1330	58-1340	58-1340	59-1350	59-1350	60-1360	60-1360	61-1375
	61-1375	62-1392	62-1392	63-1408	63-1408	64-1423	64-1423	65-1438	65-1438	66-1453
	66-1453	67-1468	67-1468	68-1485	68-1485	69-1504	69-1504	70-1514	70-1514	71-1524
	71-1524	72-1540	72-1540	72-1571	72-1571	72-1588	72-1588	73-1605	73-1605	74-1627
	74-1627	75-1673	75-1673	76-1708	76-1708	77-1761	77-1761	78-1838	78-1838	79-1876
	79-1876	80-1936	80-1936	81-1971	81-1971	82-2006	82-2006	83-2048	83-2048	84-2095
	84-2095	85-2140	85-2140	86-2187	86-2187	87-2233	87-2233	88-2270	88-2270	89-2339
	89-2339	90-2427	90-2427	91-2473	91-2473	92-2502	92-2502	93-2531	93-2531	94-2566
	94-2566	95-2604	95-2604	96-2642	96-2642	97-2680	97-2680	98-2717	98-2717	99-2772
	99-2772	100-2812	100-2812	101-2898	101-2898	102-2919	102-2919	103-2937	103-2937	104-2959
	104-2959	105-2982	105-2982	106-3032	106-3032	107-3077	107-3077	108-3127	108-3127	109-3169
	109-3169	110-3226	110-3226	111-3284	111-3284	112-3342	112-3342	113-3392	113-3392	114-3437
	114-3437	115-3488	115-3488	116-3530	116-3530	117-3587	117-3587	118-3645	118-3645	119-3703
	119-3703	120-3763	120-3763	121-3808	121-3808	122-3859	122-3859	123-3924	123-3924	124-3970
	124-3970	125-4065	125-4065	126-4162	126-4162	127-4268	127-4268	128-4375	128-4375	129-4423
	129-4423	130-4481	130-4481	131-4532	131-4532	132-4586	132-4586	133-4629	133-4629	134-4673
	134-4673	135-4718	135-4718	136-4789	136-4789	137-4861	137-4861	138-4932	138-4932	139-5002
	139-5002	140-5089	140-5089	141-5140	141-5140	142-5169	142-5169	143-5229	143-5229	144-5329
	144-5329	145-5413	145-5413	146-5469	146-5469	147-5533	147-5533	148-5629	148-5629	149-5753
	149-5753	150-5879	150-5879	151-5974	151-5974	152-6067	152-6067	153-6163	153-6163	154-6266
	154-6266	155-6309	155-6309	156-6353	156-6353	157-6393	157-6393	158-6444	158-6444	159-6501
	159-6501	160-6543	160-6543	161-6591	161-6591	162-6644	162-6644	163-6707	163-6707	164-6754
	164-6754	165-6787	165-6787	166-6863	166-6863	167-6903	167-6903	168-6991	168-6991	169-7061
	169-7061	170-7093	170-7093	171-7127	171-7127	172-7160	172-7160	173-7216	173-7216	174-7218
	175-7220	176-7222	176-7222	176-7222	177-7224	178-7226	179-7228	180-7230	181-7234	183-7452
TRMTRP	#180-7230									
TYPDEC	#13-435	172-7203								
TYPOCS	#13-435	#182-7284	#182-7393	#182-7394	#182-7396	#182-7397	#182-7399	#182-7401	#182-7403	#182-7409
	#182-7411	#182-7413	#182-7415	#182-7421	#182-7423	#182-7425				
TYPOCT	#13-435	182-7286	182-7405	182-7407	182-7417	182-7419	182-7427	182-7429		
TYPTXT	#13-434	14-553	14-554	14-555	14-556	14-572	89-2416	89-2417	167-6930	167-6931
	167-6932	172-7202	182-7272	182-7283	182-7285	182-7301	182-7304	182-7356	182-7358	182-7360
	182-7362	182-7364	182-7366	182-7368	182-7370	182-7372	182-7374	182-7376	182-7378	182-7380

