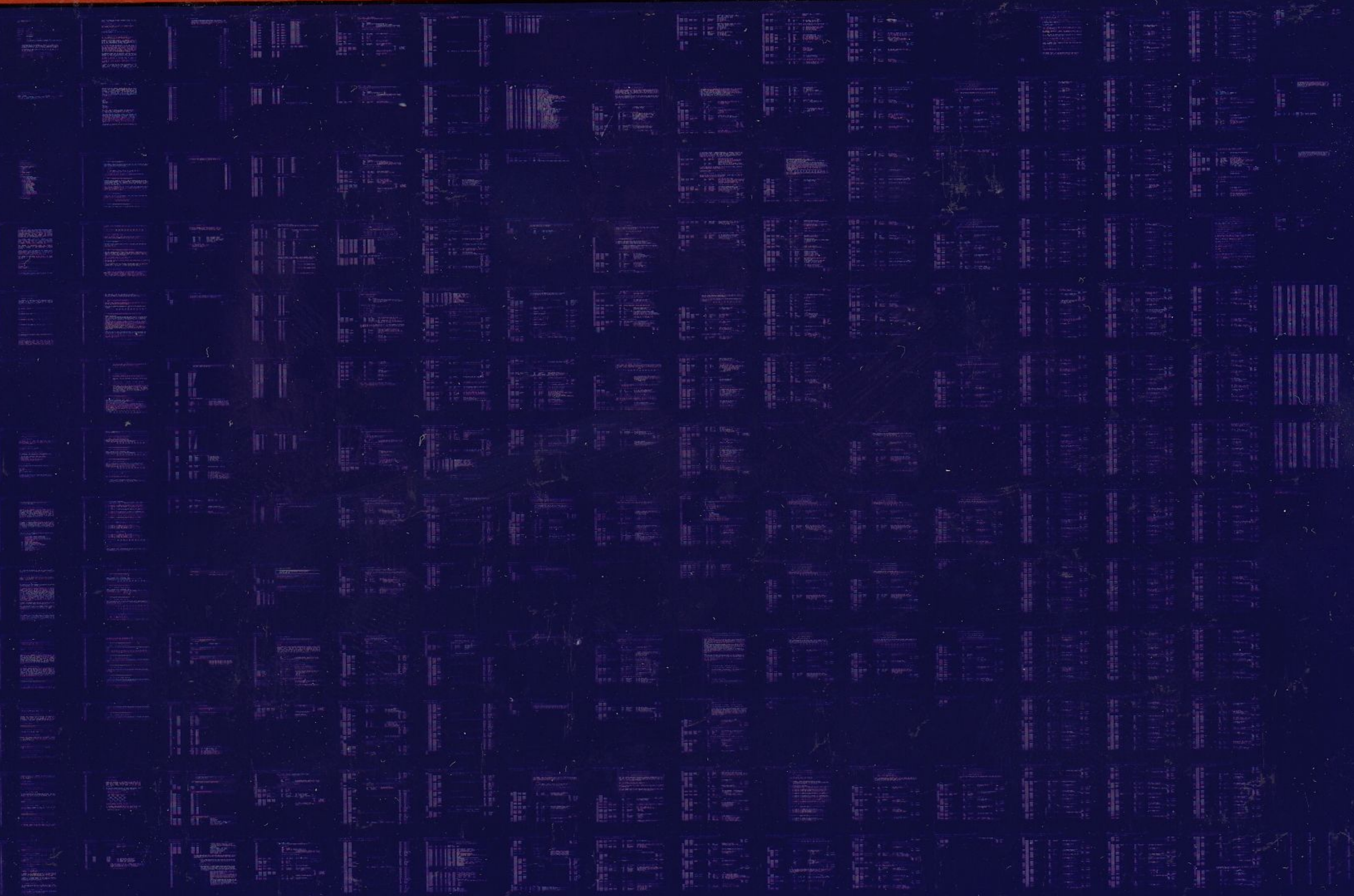


11/21+
DMV-11

DMV11 MCTAL DIAG #1
CNDMARA0

COPYRIGHT (c) 1981-84
AH-T827A-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA



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

.TITLE CNDMAAO DMV11 MCTRL DIAG #1
.SBTTL PROGRAM DOCUMENT
.REM †

I D E N T I F I C A T I O N

PRODUCT CODE: AC T826A MC
PRODUCT NAME: CNDMAAO DMV 11 MICRO CONTROLLER STATIC DIAGNOSTIC PART 1
PRODUCT DATE: APRIL 1984
MAINTAINER: ISS DIAGNOSTICS
AUTHORS: CHRIS BRIENEN
RAY MARSHALL
MODIFIED BY: JAKI BERG 9 APR 1984
PURPOSE: THIS DIAGNOSTIC IS DESIGNED TO PERFORM STATIC LOGIC TESTS FOR
THE M8053 OR M8064 (HEREAFTER REFERRED TO AS THE DMV OR DMV 11)

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 ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
AFFILIATED COMPANIES.

COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBUS
DEC DECUS DECTAPE

PROGRAM DOCUMENT

CONTENTS

63	
64	
65	
66	
67	
68	1.0 INTRODUCTION
69	
70	2.0 HARDWARE REQUIREMENTS
71	
72	3.0 PRELIMINARY PROGRAM REQUIREMENTS
73	
74	4.0 GENERAL PROGRAM CONSIDERATIONS
75	4.1 DIAGNOSTIC SUPERVISOR
76	4.2 EXECUTION TIME
77	4.3 XXDP.
78	4.4 ACT/SLIDE
79	4.5 APT
80	4.6 MEMORY MANAGEMENT
81	4.7 ERROR LOGGING
82	
83	5.0 PROGRAM LOAD MEDIA
84	
85	6.0 OPERATING INSTRUCTIONS
86	6.1 LOADING AND STARTING PROCEDURES
87	6.1.1 LOADING PROCEDURES
88	6.1.2 STARTING PROCEDURES
89	6.1.3 ** STEPS FOR QUICK AND SIMPLE EXECUTION **
90	6.2 INITIAL DIALOGUE
91	6.3 PROGRAM OPTIONS
92	6.3.1 START COMMAND
93	6.3.2 RESTART COMMAND
94	6.3.3 CONTINUE COMMAND
95	6.3.4 PROCEED COMMAND
96	6.3.5 ADD COMMAND
97	6.3.6 DROP COMMAND
98	6.3.7 PRINT COMMAND
99	6.3.8 DISPLAY COMMAND
100	6.3.9 FLAGS COMMAND
101	6.3.10 ZFLAGS COMMAND
102	6.3.11 CONTROL CHARACTERS
103	6.3.12 HARDWARE PARAMETERS
104	6.3.13 SOFTWARE PARAMETERS
105	6.3.14 EXTENDED DISCUSSION OF P TABLE DIALOGUE
106	
107	7.0 TEST DESCRIPTIONS
108	
109	8.0 ERROR INFORMATION
	8.1 ERROR REPORTING

111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166

1.0 INTRODUCTION

THE M8053 AND M8064 ARE SINGLE-LINE SYNCHRONOUS, MICRO PROCESSOR BASED COMMUNICATIONS INTERFACES WHICH CAN SUPPORT BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT ORIENTED (SDLC, MDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS TO PERFORM DIAGNOSTIC TESTING OF THE CSRS, RAM, AND BASIC MICRO-PROCESSOR LOGIC ON THESE BOARDS. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: DMV RESIDENT U-DIAG EXECUTION CSR ADDRESSING, VIA REGISTER STATIC BIT INTERACTION AND READ/WRITE TESTING, AND ON-BOARD RAM TESTING.

THE STATIC LOGIC TESTS WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO "LOCK" ONTO INTERMITTENT ERRORS. IN ADDITION TESTS ARE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

THIS PROGRAM IS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN CONFORMS TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM IS COMPATIBLE WITH ACT, APT, XXDP., AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM ALLOWS MODIFICATION OF DEVICE PARAMETERS, SUCH AS LSI BUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8053/8064 STATIC LOGIC TESTS:

SBC-11/21.
16K WORDS OF MEMORY
CONSOLE TERMINAL
M8053 OR M8064 COMMUNICATIONS INTERFACE

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM (CNDMA) SHOULD BE THE FIRST OF THE FIVE DMV 11 STATIC DIAGNOSTICS TO BE RUN. ERRORS FOUND IN THIS PROGRAM SHOULD BE CORRECTED BEFORE RUNNING THE OTHERS.

PROGRAM DOCUMENT

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
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

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE MAXIMUM TIME REQUIRED TO RUN THIS PROGRAM PER PASS FOR EACH UNIT IS 160 SECS.

4.3 XXDP.

THIS PROGRAM MAY BE LOADED UNDER XXDP., AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.

4.7 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM ANY MEDIA SUPPORTED BY XXDP.. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP., THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY

G1

CNDMA90 DMV11 MCTRL DIAG #1

MACRO M1200 22-FEB 84 15:22 PAGE 6 1

PROGRAM DOCUMENT

SEQ 0006

225

THE DIAGNOSTIC PROGRAM.

PROGRAM DOCUMENT

227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP* LOAD MEDIA. WHEN LOADED UNDER XXDP*, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP*, WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS C>)
- C) ENTER STA<CR>
- D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
- E) GET END OF PASS MESSAGES OR ERROR MESSAGES
- F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

```
DRS LOADED
DIAG. RUN TIME SERVICES
CNDMA A 0
DMV 11 U CONTRL LOGIC DIAG PART 1 OF 2
UNIT IS M8053 OR M8064
DR>
```

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

PROGRAM DOCUMENT

284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340

STA(R1)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/EOP:<INCR>

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

- MOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
- LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
- IER INHIBIT ERROR REPORTING
- IBE INHIBIT BASIC ERROR REPORTS
- IXE INHIBIT EXTENDED ERROR REPORTS
- PRI DIRECT ALL MESSAGES TO A LINE PRINTER
- PNT PRINT NUMBER OF TEST BEING EXECUTED
- BOE BELL ON ERROR
- UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
- ISR INHIBIT STATISTICAL REPORTS
- IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
- LOT LOOP ON TEST

341 THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0
342 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS
343 SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT
344 END OF 6.3.1.5.
345

346
347
348 6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

349 <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
350 PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE
351 PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE
352 EXAMPLE AT END OF 6.3.1.5.
353

354
355 6.3.1.5 EFFECT OF START COMMAND
356

357 THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE
358 PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND
359 THEN THE DIAGNOSTIC TESTS THEMSELVES.
360

361 THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
362 "# UNITS?" TO WHICH THE OPERATOR REPLIES WITH A DECIMAL
363 NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE DEVICE
364 TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING
365 THIS ARE THE QUESTIONS WHEREBY THE P TABLES THEMSELVES WILL
366 BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING
367 ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR
368 MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION.
369 HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN
370 WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
371 BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION
372 (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY
373 THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR
374 OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE
375 AFTER THE PARENTHESES.
376

377 FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS
378 TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK
379 VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.
380

381 WHEN THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS
382 ALLOCATED FOR THE P TABLES, AND IF THERE IS NOT ENOUGH TO
383 ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN
384 THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO
385 TEST ALL UNITS.
386

387 EXAMPLE:
388

389 STA/TESTS:1:2 4:6:8 10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE
390

391 THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS
392 CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST
393 ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND
394 SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON
395 A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS
396 PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST
397 THREE LETTERS ARE SCANNED.

PROGRAM DOCUMENT

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
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454

6.3.2 RESTART COMMAND

RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/UNITS:<UNIT LIST>

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS CNT>, AND <FLAG-LIST> ARE AS IN THE START
COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR
RANGES OF DECIMAL NUMBERS (0-5, 8 10 ETC.) THAT SPECIFY THE
UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS.
THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF
UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER
INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS
ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE
SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND.
SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT
IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP
COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT
THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST
HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT.
THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF
THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED
(OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER
COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL
WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B)
AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C)
A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

CON(TINUE)/PASS:<PASS CNT>/FLAGS:<FLAG LIST>

6.3.3.1 PASS SWITCH (/PASS:<PASS CNT>)

<PASS CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS
THE UNSATISFIED PASS CNT FROM THE PREVIOUS START OR RESTART.
IF NONE REMAINS, THE DEFAULT IS NON ENDING EXECUTION.

PROGRAM DOCUMENT

455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED
FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE
MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A
CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE
BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT
OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY
BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

PRO(CEED)/FLAGS:<FLAG-LIST>

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED
FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND
MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT
OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION
FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE
PARAMETERS MAY BE ALTERED.

6.3.5 ADD COMMAND

ADD/UNITS:<UNIT LIST>

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH
UNIT MUST HAVE A P TABLE IN MEMORY DUE TO AN EARLIER

PROGRAM DOCUMENT

512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568

HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

DRO(P)/UNITS:<UNIT-LIST>

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

6.3.8 DISPLAY COMMAND

DIS(PLAY)/UNITS:<UNIT-LIST>

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT LIST>)

<UNIT LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO

PROGRAM DOCUMENT

569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625

DESIGNATED.

6.3.9 FLAGS COMMAND

FLA(GS)

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

ZFL(AGS)

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- HARD CORE QUESTIONS (SEE 6.2), HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 3 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE

1. DEVICE CSR ADDRESS : (O) 160020?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE LSI-BUS. THE ALLOWABLE RANGE IS 160020 177760 (OCTAL), AND THE DEFAULT VALUE IS 160020.

2. DEVICE VECTOR ADDRESS : (O) 300 ?

626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

3. DEVICE PRIORITY LEVEL : (0) 4 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0 7, AND THE DEFAULT VALUE IS 4.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 OF THE STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "N UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6 10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO

PROGRAM DOCUMENT

683 CONSTRUCT A SET OF P TABLES. ASSUME THAT WE HAVE 16 UNITS,
684 AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE
685 SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE
686 DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE
687 THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR
688 THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER
689 (0,1,2,.....15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE
690 VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE
691 THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR
692 THE LAST 9 UNITS.
693
694 THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:
695
696 @ UNITS (D) ? 16
697 UNIT 0
698 <QUESTION 1> ? 75
699 <QUESTION 2> ? 0-6
700 <QUESTION 3> ? 76
701
702 UNIT 7
703 <QUESTION 1> ?
704 <QUESTION 2> ? 7-11,13 15
705 <QUESTION 3> ? 77
706
707 THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75
708 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,.....6
709 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15.
710 SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.
711
712 THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE
713 GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS
714 PRINTED OUT FOR THE OPERATOR IN THE FORM "UNIT XX" AT
715 THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO
716 BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7
717 THRU 15. SINCE NO NEW EXPLICIT VALUES ARE TYPED IN, SLOT TWO
718 GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND
719 GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN
720 TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7
721 THRU 15.
722
723 THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT
724 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION
725 (NAMELY QUESTION 2).
726
727

PROGRAM DOCUMENT

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
775
776
777
778
779
780
781
782
783
784
785

7.0 TEST DESCRIPTIONS

```

;*****
;*      TEST 1 <DMV-11 AVAILABILITY>
;*
;*  EACH NORMALLY USED CSR IS ACCESSED WITH A "TST" OR "TSTB" INSTRUCTION AND IF
;*  A BUS TIMEOUT OCCURS (INTERRUPT @ VECTOR ADDR 4) A FLAG WILL BE SET SHOWING
;*  WHICH CSR ADDR AND INSTRUCTION FAILED. "T1.HSW" REFLECTS "TST" INSTRUCTIONS
;*  AND "T1.HSB" REFLECTS "TSTB" INSTRUCTIONS.
;*
;*  EXAMPLES:
;*
;*      IF "TSTB @BSEL1" FAILS, BIT # 1 OF "T1.HSB" WILL BE SET.
;*      IF "TST @SEL4" FAILS, BIT # 4 OF "T1.HSW" WILL BE SET
;*      (NOTE: ONLY EVEN BITS IN "T1.HSW" CAN BE SET).
;*
;*  THE FLAG WORDS ARE OUTPUT IN BINARY AS "EXTENDED ERROR INFORMATION".
;*****

```

```

;*****
;*      TEST 2 <MASTER CLEAR, RUN MICRODIAGNOSTICS>
;*
;*  A MASTER CLEAR IS ISSUED TO THE DMV-11, AND THE PROGRAM ALLOWS SUFFICIENT
;*  TIME FOR THE MICRODIAGNOSTICS TO BE PERFORMED. THESE MICRODIAGNOSTIC RESIDE
;*  IN 6502 PROGRAM MEMORY, AND THOROUGHLY VERIFY THE OPERATION OF THE 6502
;*  MICROPROCESSOR CHIP. THEN, THEY CHECK OUT THE DATA RAM, THE 6502'S ACCESS TO
;*  THE CSR'S, AND PERFORM A SIMPLE MESSAGE TEST USING THE 6522 CHIP AND THE
;*  USYRT, WITH INTERNAL LOOPBACK.
;*
;*  NEXT, THE LSI-11 PROGRAM READS THE THE CSR'S (SELO SEL6) AND CHECKS THEM FOR
;*  THEIR EXPECTED INITIALIZED STATES. IF AN ERROR HAS OCCURRED IN THE MICRO
;*  DIAGNOSTICS THE NUMBER OF THE FAILING TEST WILL BE FOUND IN SEL4, AND RUN
;*  (BIT 7) WILL NOT BE SET IN BSEL1.
;*****

```

```

;*****
;*      TEST 3 <CSR ADDRESSING>
;*
;*  FIRST, HALT THE 6502 UP BY CLEARING ALL CSRS. THEN, WRITE A DIFFERENT WORD
;*  OF DATA PATTERN A INTO EACH OF BSEL0-17, AND AFTER EACH WRITE, READ AND
;*  COMPARE ALL REGS TO EXPECTED VALUES.
;*
;*  DATA PATTERN A = 001, 002, 004, 010, 020, 040, 100, 200, 052, 300, 140,
;*                   060, 030, 014, 006, 003
;*****

```

```

;*****
;*      TEST 4 <CSR REGISTERS DATA READ/WRITE>
;*
;*  WRITE, READ, AND COMPARE EACH BYTE OF DATA PATTERN B INTO REGISTER BSEL0.
;*  THEN, REPEAT THIS USING EACH OF THE REMAINING CSR'S, BSEL1 BSEL1'. WHEN BSEL1

```

PROGRAM DOCUMENT

786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842

```

;* IS BEING TESTED, THE PROGRAM ALWAYS SETS BIT 7 IN THE DATA PATTERN SO THAT
;* RUN WILL NOT BE CLEARED, AND IT ALWAYS CLEARS BIT6 SO THAT MCLR WILL NOT BE
;* SET.
;*
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
;*****

;*****
;* TEST 5 <BASIC MASTER CLEAR>
;*
;* PERFORM AN INITIAL MASTER CLEAR, WRITE 356 INTO BSELO AND READ AND CHECK IT.
;* THEN, ISSUE A MASTER CLEAR AND READ AND CHECK BSELO FOR 000.
;*****

;*****
;* TEST 6 <BUS RESET>
;*
;* PERFORM AN INITIAL MASTER CLEAR. WRITE 377 INTO BSELO AND READ AND CHECK
;* IT. THEN, ISSUE A RESET INSTRUCTION, STALL FOR COMPLETION, AND READ AND
;* CHECK BSELO FOR 000.
;*****

;*****
;* TEST 7 <CSR, MAINTENANCE MICROCODE INTERACTION>
;*
;* THIS TEST INVOKES THE MAINTENANCE REQUEST MECHANISM THROUGH WHICH THE LSI-11
;* AND 6502 CAN COMMUNICATE. FIRST, A MASTER CLEAR IS DONE WITH ONLY BIT 0
;* (MREQ) SET IN BSEL1. THE PROGRAM THEN CHECKS FOR THE SETTING OF BSEL2 BIT 7
;* (MRDY) BY THE MAINTENANCE MICROCODE WITHIN ABOUT 50 MICRO SEC., AND IF MRDY
;* DOES NOT GET SET, AN ERROR IS REPORTED.
;*
;* NEXT, THE PROGRAM LOADS SEL4 WITH 000010 AND BSEL6 WITH 125. THEN, ALL CSR'S
;* ARE READ AND CHECKED FOR EXPECTED CONTENTS.
;*
;* BSEL2 IS THEN LOADED WITH A WRITE COMMAND, WHICH SHOULD CAUSE THE MICROCODE
;* TO TRANSFER THE 125 INTO BSELO. ALL CSR'S ARE THEN READ AND CHECKED FOR
;* EXPECTED CONTENTS.
;*
;* THEN, THE PROGRAM LOADS 252 INTO BSELO AND READS AND CHECKS ALL CSR'S. BSEL2
;* IS THEN LOADED WITH A READ COMMAND, WHICH SHOULD CAUSE THE MICROCODE TO
;* TRANSFER THE 252 INTO BSEL6. ALL CSR'S ARE READ AND CHECKED.
;*****

;*****
;* TEST 8 <RUN FLIP-FLOP>
;*
;* THE PROGRAM PUTS THE MICROCODE INTO THE MAINTENANCE LOOP. A 125 CHARACTER
;* IS LOADED INTO BSEL6 AND A REQUEST IS MADE TO WRITE THE CONTENTS OF BSEL6
;* INTO BSELO. THE PROGRAM THEN READS AND CHECKS BSELO TO CONTAIN 125.
;* NEXT, THE RUN FLIP-FLOP IS CLEARED BY LOADING A 0 INTO RUN (BSEL1 BIT 7).
;* BSELO IS THEN CLEARED AND THE REQUEST IS MADE AGAIN TO WRITE THE CONTENTS
;* OF BSEL6 INTO BSELO. THE PROGRAM STALLS FOR 50 MICRO SEC. AND CHECKS FOR

```

PROGRAM DOCUMENT

```

843 ;* MRDY (BSEL2 BIT 7) NOT SET, AND BSEL0 STILL CLEARED.
844 ;* THEN, THE PROGRAM SETS THE RUN FLIP-FLOP AGAIN BY LOADING A 1 INTO RUN.
845 ;* AND CHECKS FOR MRDY SET WITHIN 50 MICRO-SEC. AND BSEL0 = 125.
846 ;*****
847
848
849 ;*****
850 ;* TEST 9 <LOW RAM (00-0F) SCRATCHPAD>
851 ;*
852 ;* THIS TEST FIRST PERFORMS AN ADDRESSING TEST OF RAM LOCATIONS (00-0F), BY
853 ;* WRITING THE ADRS INTO EACH LOCATION AND AFTER EACH WRITE, ALL THE LOCATIONS
854 ;* ARE READ AND CHECKED FOR EXPECTED CONTENTS.
855 ;*
856 ;* THEN, THE TEST PERFORMS READ/WRITE DATA TESTING OF RAM LOCATIONS 00 OF,
857 ;* BY WRITING, READING, AND COMPARING ALL BYTES OF DATA PATTERN B IN EACH
858 ;* LOCATION.
859 ;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
860 ;* 200, 376, 375, 373, 367, 357, 337, 277, 177, 000
861 ;*****
862
863
864 ;*****
865 ;* TEST 10 <DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)>
866 ;*
867 ;* GENERAL DESCRIPTION:
868 ;* FIRST, THE 2K BYTE LOCATIONS IN RAM ARE LOADED WITH 0'S (SEE NOTE BELOW).
869 ;* THEN, THE FIRST LOCATION IS READ AND CHECKED, A SINGLE 1 IS WRITTEN INTO
870 ;* THE LOW BIT POSITION, AND THIS IS READ AND CHECKED. THIS IS DONE FOR ALL
871 ;* BYTES IN THE RAM, BY INCREMENTING THE ADDRESS TO POINT TO THE NEXT RAM
872 ;* LOCATION.
873 ;* THEN, THE NEXT BIT POSITION IS CHOSEN TO INSERT A 1, AND ALL LOCATIONS
874 ;* ARE READ, WRITTEN, AND READ AS BEFORE. THIS IS CONTINUED FOR ALL BIT
875 ;* POSITIONS UNTIL THE ENTIRE RAM IS WRITTEN TO ALL 1'S. THE ABOVE OPERATIONS
876 ;* ARE PERFORMED A SECOND TIME, WITH 0'S INSERTED INTO THE RAM INSTEAD OF 1 S.
877 ;* THIS RESULTS IN THE ENTIRE RAM BEING WRITTEN TO ALL 0'S.
878 ;* THIS TEST CONSTITUTES A THOROUGH TEST OF THE RAM. IT IS CAPABLE OF
879 ;* DETECTING THE FOLLOWING FAULTS : STUCK ADDRESS BITS, UNI- AND BI-DIRECT
880 ;* IONAL COUPLING BETWEEN ADDRESS BITS, STUCK MEMORY BITS, AND UNI- AND
881 ;* BI-DIRECTIONAL COUPLING BETWEEN MEMORY BITS IN BOTH ROWS AND COLUMNS OF THE
882 ;* MEMORY MATRIX.
883 ;*
884 ;* NOTE:
885 ;* THIS TEST DOES NOT CHECK LOCATIONS 0010-001F, SO THAT THE PRIMARY CSR'S
886 ;* ARE NOT WRITTEN. IT DOES TEST LOCATIONS 0000-000F (SCRATCHPAD RAM) AND
887 ;* LOCATIONS 0020-002F (SECONDARY CSR'S), AS WELL AS 0030-0800 (BASIC RAM).
888 ;*
889 ;* THE "TMP#" REGISTERS ARE USED HERE TO CONTAIN THE VARIOUS CONSTANTS &
890 ;* VARIABLES USED THROUGHOUT THIS TEST. A LIST OF THEIR ASSIGNMENTS SEEMS
891 ;* USEFUL SO HERE IT IS:
892 ;*
893 ;* TMP0 POINTS TO THE FIRST LOCATION AFTER THE SELECT REGISTERS.
894 ;*
895 ;* TMP1 - - -
896 ;*
897 ;* TMP2 TEST PATTERN ID CODE UNUSED BY THIS TEST.
898 ;*
899 ;* TMP3 TEST DATA PATTERN INDEX UNUSED BY THIS TEST.

```

PROGRAM DOCUMENT

```

900      ;*
901      ;*      TMP4      TEST DATA PATTERN.  THE HIGH BYTE IS THE PATTERN BEING WRITTEN
902      ;*      ON ANY GIVEN PASS AND THE LOW BYTE IS THE PATTERN THAT WAS
903      ;*      WRITTEN BY THE PREVIOUS PASS THROUGH THE RAM.
904      ;*
905      ;*      TMP5      DATA READ FROM THE RAM.  ONLY THE LOW BYTE IS USED.
906      ;*
907      ;*      TMP6      --
908      ;*      TMP7      ----
909      ;*      TMP8      ---
910      ;*      TMP9      .
911      ;*
912      ;*      TMPA      RAM ADDRESS BEING TESTED.
913      ;*
914      ;*      TMPB      BIT POINTER.  NUMBER OF THE BIT WITHIN THE DATA FIELD WHICH IS
915      ;*      BEING SWITCHED ON EACH WRITE WITHIN THE CURRENT PASS.
916      ;*
917      ;*      TMPC      DATA FLAG.  BIT 0 OF THIS WORD IS THE VALUE TO WHICH THE BIT
918      ;*      IDENTIFIED IN TMPB IS BEING SET ON EACH WRITE IN THE CURRENT
919      ;*      PASS.
920      ;*
921      ;*      TMPD      DIRECTION SWITCH.  0 = FORWARD      NON ZERO = BACKWARD
922      ;*
923      ;*      TMPE      LAST VALID ADDRESS TO BE TESTED.  (I.E. THE END OF RAM)
924      ;*
925      ;*      TMPF      ERROR FLAGS.  BIT 1 SET = THE LAST DETECTED ERROR WAS THE READ
926      ;*      OF THE PREVIOUS DATA BEFORE WRITING THE NEW DATA.  IF BIT2 IS
927      ;*      SET, THE READ AFTER WRITE FAILED.  IF EITHER IS SET WHEN AN
928      ;*      ERROR IS DETECTED, THE SUPERVISOR IS NOT CALL'D AND THEREFOR
929      ;*      IT'S ERROR COUNTER WILL NOT REFLECT THE ERROR - INSTEAD, THE
930      ;*      DATA LINE IS PRINTED.  (UNLESS THE ERROR HANDLER'S DATA LINE
931      ;*      PRINT COUNT HAS EXCEEDED ITS LIMIT      IN WHICH CASE ITS
932      ;*      INVOCATION IS IGNORED.)
933      ;*
934      ;*****
935
936      ;*****
937      ;*      TEST 11 <VIA REGISTER ADDRESSING>
938      ;*
939      ;*      VIA == "6522 VERSATILE INTERFACE ADAPTER"
940      ;*
941      ;*      A MASTER CLEAR IS PERFORMED, NEXT, TIMER 1 LATCHES
942      ;*      ARE CLEARED BY WRITING 000 INTO VIA REGS 6 & 7
943      ;*      THEN, 377 IS LOADED INTO DATA DIRECTION REGISTERS A, B (DDRA, DDRB) TO
944      ;*      SET THE PORT PINS FOR OUTPUT MODE.
945      ;*      THEN, A DIFFERENT BYTE OF DATA PATTERN C IS WRITTEN INTO EACH VIA
946      ;*      LOCATION, (EXCEPT THE TIMER REGS 4,5,10,11 OCT) AND AFTER EACH IS WRITTEN,
947      ;*      ALL VIA REGS (EXCEPT 4,5,10,11) ARE READ AND COMPARED TO EXPECTED
948      ;*      CONTENTS.  NOTE THAT SOME VIA REGS ARE ALTERED BY THE LOADING OF OTHERS,
949      ;*      AND THE PROGRAM TAKES THIS INTO ACCOUNT, IN THE SETTING OF EXPECTED REG
950      ;*      VALUES.  THE DATA PATTERN IS CHOSEN TO AVOID ACTIVATING THE VIA CHIP (SUCH
951      ;*      AS GENERATING OUTPUTS ON CA1, CA2, CB1, CB2, OR CAUSING 6502
952      ;*      INTERRUPT REQUESTS).
953      ;*      DATA PATTERN C (WITH VIA REGS AND THEIR DATA SHOWN IN OCTAL) :
954      ;*      REGISTER = 00  01  02  03  06  07  12  13  14  15  16  17
955      ;*      DATA = 100, 101, 377, 377, 106, 107, 112, 040, 042, 000, 200, 117
956      ;*      NEXT, 000 IS LOADED INTO DDRA, AND DDRB IS READ AND COMPARED TO 377.  THEN,

```

PROGRAM DOCUMENT

```

957 ;* THE 377 IS LOADED BACK INTO DDRA, AND DDRB IS LOADED WITH 000 AND DDRA IS
958 ;* READ AND COMPARED TO 377.
959 ;*****
960
961
962 ;*****
963 ;* TEST 12 <VIA'S DDRB DATA READ/WRITE>
964 ;*
965 ;* DDRB == "DATA DIRECTION REGISTER B"
966 ;* FIRST, A MASTER CLEAR IS PERFORMED. THEN :
967 ;* READ/WRITE BITS 0-7 OF VIA DATA DIRECTION REGISTER B ARE TESTED BY WRITING,
968 ;* READING, AND COMPARING EACH BYTE OF DATA PATTERN B.
969 ;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
970 ;* 200, 376, 375, 373, 367, 357, 337, 277, 177, 000
971 ;*****
972
973
974 ;*****
975 ;* TEST 13 <VIA'S DDRA DATA READ/WRITE>
976 ;*
977 ;* DDRA == "DATA DIRECTION REGISTER A"
978 ;*
979 ;* FIRST, A MASTER CLEAR IS PERFORMED. THEN :
980 ;* READ/WRITE BITS 0-7 OF VIA DATA DIRECTION REGISTER A ARE TESTED BY WRITING,
981 ;* READING, AND COMPARING EACH BYTE OF DATA PATTERN B.
982 ;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
983 ;* 200, 376, 375, 373, 367, 357, 337, 277, 177, 000
984 ;*****
985
986
987 ;*****
988 ;* TEST 14 <VIA'S ORB DATA READ/WRITE>
989 ;*
990 ;* ORB == "OUTPUT REGISTER PORT B"
991 ;*
992 ;* FIRST, A MASTER CLEAR IS PERFORMED. NEXT, 377 IS LOADED INTO DATA
993 ;* DIR. REG. B (DDRB) TO SET ALL B PORT PINS FOR OUTPUT MODE. THEN ,
994 ;* READ/WRITE BITS 0-7 OF VIA OUTPUT REG. PORT B ARE TESTED BY WRITING,
995 ;* READING, AND COMPARING EACH BYTE OF DATA PATTERN B.
996 ;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
997 ;* 200, 376, 375, 373, 367, 357, 337, 277, 177, 000
998 ;*****
999
1000
1001 ;*****
1002 ;* TEST 15 <VIA'S T1 DATA READ/WRITE>
1003 ;*
1004 ;* T1 == 'TIMER #1"
1005 ;*
1006 ;* THIS TEST WRITES, READS, AND CHECKS THE T1 LATCHES AND COUNTER REGISTERS
1007 ;* WITH DATA PATTERNS IN EACH OF 3 SUBTESTS.
1008 ;*
1009 ;*
1010 ;* FIRST SUBTEST: CHECKS FOR PROPER LOADING OF LATCHES
1011 ;* IT ALSO CHECKS TO BE SURE THAT THE COUNTER APPEARS TO BE DOING
1012 ;* SOMETHING TO THE COUNTERS. AS LONG AS THEY HAVE CHANGED FROM THE
1013 ;* VALUE LOADED INTO THEM, WE WILL BE SATISFIED.

```

PROGRAM DOCUMENT

```

1014 ;*
1015 ;* A. A MASTER CLEAR IS PERFORMED.
1016 ;* B. ALL LATCHES ARE LOADED TO ZEROES (JUST IN CASE), ACR6 & ACR7 ARE SET
1017 ;* TO ZERO (MODE 00), AND "T1" INTERRUPT ENABLE FLAG IS CLEARED.
1018 ;*
1019 ;* C. T1L-L(ADR 04) IS LOADED WITH THE CURRENT BYTE OF DATA PATTERN B.
1020 ;* D. T1L-L(ADR 06) IS READ AND COMPARED TO THE BYTE JUST WRITTEN.
1021 ;* E. T1C-L(ADR 04) IS READ AND CHECKED TO BE DIFFERENT THAN THE TEST BYTE.
1022 ;*
1023 ;* F. T1L-L(ADR 06) IS LOADED WITH THE COMPLEMENT OF THE CURRENT DATA BYTE.
1024 ;* G. T1L-L(ADR 06) IS READ AND COMPARED TO THE BYTE JUST WRITTEN.
1025 ;*
1026 ;* H. T1L-L(ADR 06) IS RE-LOADED WITH 0 TO MAKE T1C-H DECREMENT FAST.
1027 ;* T1L-H(ADR 05) IS LOADED WITH THE ORIGINAL TEST DATA PATTERN BYTE.
1028 ;* I. T1L-H(ADR 07) IS READ AND COMPARED TO THE BYTE LOADED INTO T1L-H.
1029 ;*
1030 ;* J. T1C-H(ADR 05) IS READ AND CHECKED TO BE DIFFERENT THAN THE TEST BYTE.
1031 ;*
1032 ;* K. T1L-H(ADR 07) IS LOADED WITH THE COMPLEMENT OF THE CURRENT DATA BYTE.
1033 ;* L. T1L-H(ADR 07) IS READ AND COMPARED TO THE BYTE JUST LOADED.
1034 ;*
1035 ;* M. STEPS C-L ARE REPEATED USING EACH BYTE OF DATA PATTERN B.
1036 ;*
1037 ;*
1038 ;* SECOND SUBTEST: CHECKS FOR CROSS-TALK AND ADDRESSING ERRORS
1039 ;* FROM T1L-L TO T1L-H
1040 ;*
1041 ;* A. T1L-H(ADR 07) IS LOADED WITH 000 TO CLEAR IT.
1042 ;* B. T1L-L(ADR 06) IS LOADED WITH A BYTE OF DATA PATTERN B.
1043 ;* C. T1L-L(ADR 06) IS READ AND COMPARED TO THE DATA JUST WRITTEN.
1044 ;* D. T1L-H(ADR 07) IS READ AND COMPARED TO 000.
1045 ;* E. STEPS B-D ARE REPEATED USING EACH BYTE OF DATA PATTERN B.
1046 ;*
1047 ;*
1048 ;* THIRD SUBTEST: CHECKS FOR CROSS TALK AND ADDRESSING ERRORS
1049 ;* FROM T1L-H TO T1L L
1050 ;*
1051 ;* A. T1L-L(ADR 04) IS LOADED WITH 000 TO CLEAR IT
1052 ;* B. T1L-H(ADR 07) IS LOADED WITH A BYTE OF DATA PATTERN B.
1053 ;* C. T1L H(ADR 07) IS READ AND COMPARED TO THE DATA JUST WRITTEN.
1054 ;* D. T1L-L(ADR 06) IS READ AND COMPARED TO 000.
1055 ;* E. STEPS B-D ARE REPEATED USING EACH BYTE OF DATA PATTERN B.
1056 ;*
1057 ;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
1058 ;* 200, 376, 375, 373, 367, 357, 337, 277, 177, 000
1059 ;*
1060 ;*
1061 ;*
1062 ;*
1063 ;* *****
1064 ;* TEST 16 <VIA'S SR DATA READ/WRITE>
1065 ;*
1066 ;* SR == "SHIFT REGISTER"
1067 ;*
1068 ;* FIRST, A MASTER CLEAR IS PERFORMED AND THE ACR IS SET TO 000. THEN :
1069 ;* READ/WRITE BITS 0-7 OF VIA SHIFT REGISTER ARE TESTED BY WRITING, READING,
1070 ;* AND COMPARING EACH BYTE OF DATA PATTERN B.
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,

```

PROGRAM DOCUMENT

```

1071 ;*          200, 376, 375, 373, 367, 357, 337, 277, 177, 000
1072 ;*****
1073
1074
1075 ;*****
1076 ;*      TEST 17 <VIA'S ACR DATA READ/WRITE>
1077 ;*
1078 ;*      ACR == "AUXILIARY CONTROL REGISTER"
1079 ;*
1080 ;* FIRST, A MASTER CLEAR IS PERFORMED. THEN :
1081 ;* READ/WRITE BITS 0-7 OF THE ACR ARE TESTED BY WRITING, READING,
1082 ;* AND COMPARING EACH BYTE OF DATA PATTERN B.
1083 ;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
1084 ;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
1085 ;*****
1086
1087
1088 ;*****
1089 ;*      TEST 18 <VIA'S PCR DATA READ/WRITE>
1090 ;*
1091 ;*      PCR == "PERIPHERAL CONTROL REGISTER"
1092 ;*
1093 ;* FIRST, A MASTER CLEAR IS PERFORMED. THEN :
1094 ;* READ/WRITE BITS 0-7 OF THE PCR REGISTER ARE TESTED BY WRITING, READING,
1095 ;* AND COMPARING EACH BYTE OF A SUBSET OF DATA PATTERN B.
1096 ;* DATA PATTERN B (SUBSET) = 125, 252, 000, 377, 001, 002, 004, 010, 020,
1097 ;*                           040, 100, 200.
1098 ;*****
1099
1100
1101 ;*****
1102 ;*      TEST 19 <VIA'S IER DATA READ/WRITE>
1103 ;*
1104 ;*      IER == "INTERRUPT ENABLE REGISTER"
1105 ;*
1106 ;* BITS 0-6 OF THE IER CAN BE SET OR CLEARED ON A WRITE, UNDER CONTROL OF THE
1107 ;* SET/CLEAR CONTROL BIT 7. TO TEST THIS , EACH BYTE OF DATA PATTERN D IS
1108 ;* WRITTEN INTO IER, AND THE REGISTER IS READ AND COMPARED TO THE CORRESPOND-
1109 ;* ING BYTE OF DATA PATTERN E.
1110 ;*
1111 ;* DATA PATTERN D = 200, 201, 202, 204, 210, 220, 240, 300, 200, 000, 001,
1112 ;*                  002, 004, 010, 020, 040, 100, 000, 325, 125, 252, 052
1113 ;*
1114 ;* DATA PATTERN E = 000, 001, 003, 007, 017, 037, 077, 177, 177, 177, 176,
1115 ;*                  174, 170, 160, 140, 100, 000, 000, 125, 000, 052, 000
1116 ;*
1117 ;*****
1118
1119
1120 ;*****
1121 ;*      TEST 20 <VIA'S ORB/DRB MASTER CLEAR TEST>
1122 ;*
1123 ;*      ORB == "OUTPUT REGISTER PORT B"
1124 ;*      DRB == "DATA DIRECTION REGISTER B"
1125 ;*
1126 ;* FIRST, A MASTER CLEAR IS PERFORMED. NEXT, 377 IS LOADED INTO DRB TO SET
1127 ;* ALL B PORT PINS FOR OUTPUT MODE. THEN, A 000 BYTE IS WRITTEN INTO ORB AND

```

PROGRAM DOCUMENT

```

1128 ;* THE REGISTER IS READ BACK AND CHECKED FOR 000. THEN, A MASTER CLEAR IS
1129 ;* PERFORMED AND ORB IS READ AND CHECKED FOR 377.
1130 ;*****
1131
1132
1133 ;*****
1134 ;* TEST 21 <VIA'S DDRB MASTER CLEAR TEST>
1135 ;*
1136 ;* DDRB == "DATA DIRECTION REGISTER B"
1137 ;*
1138 ;* A 377 BYTE IS WRITTEN INTO DDRB AND THE REGISTER IS READ BACK AND CHECKED
1139 ;* FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND DDRB IS READ AND CHECKED FOR
1140 ;* 000.
1141 ;*
1142 ;* NOTE: THIS TESTING IS ALSO DONE IN TEST 23. IT IS INCLUDED HERE ONLY TO
1143 ;* PROVIDE TIGHTER LOOPING ON JUST THE DDRB MASTER CLEAR CHECKING.
1144 ;*****
1145
1146
1147 ;*****
1148 ;* TEST 22 <VIA'S DDRA MASTER CLEAR TEST>
1149 ;*
1150 ;* DDRA == "DATA DIRECTION REGISTER A"
1151 ;*
1152 ;* A 377 BYTE IS WRITTEN INTO DDRA AND THE REGISTER IS READ BACK AND CHECKED
1153 ;* FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND DDRA IS READ AND CHECKED FOR
1154 ;* 000.
1155 ;*****
1156
1157
1158 ;*****
1159 ;* TEST 23 <VIA'S SR MASTER CLEAR TEST>
1160 ;*
1161 ;* SR == "SHIFT REGISTER"
1162 ;*
1163 ;* A 123 BYTE IS WRITTEN INTO SR AND THE REGISTER IS READ BACK AND CHECKED
1164 ;* FOR 123. THEN, A MASTER CLEAR IS PERFORMED AND SR IS READ AND CHECKED FOR
1165 ;* NO CHANGE.
1166 ;*****
1167
1168
1169 ;*****
1170 ;* TEST 24 <VIA'S ACR MASTER CLEAR TEST>
1171 ;*
1172 ;* ACR == "AUXILIARY CONTROL REGISTER"
1173 ;*
1174 ;* A 252 BYTE IS WRITTEN INTO ACR AND THE REGISTER IS READ BACK AND CHECKED
1175 ;* FOR 252. THEN, A MASTER CLEAR IS PERFORMED AND ACR IS READ AND CHECKED FOR
1176 ;* 000. TO VERIFY THAT IT IS CLEARED BY MASTER CLEAR.
1177 ;*****
1178
1179
1180 ;*****
1181 ;* TEST 25 <VIA'S PCR MASTER CLEAR TEST>
1182 ;*
1183 ;* PCR == "PERIPHERAL CONTROL REGISTER"
1184 ;*

```


PROGRAM DOCUMENT

```

1185      ;* A 377 BYTE IS WRITTEN INTO PCR AND THE REGISTER IS READ BACK AND CHECKED
1186      ;* FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND PCR IS READ AND CHECKED FOR
1187      ;* 000.
1188      ;*****
1189
1190
1191      ;*****
1192      ;*      TEST 26 <VIA'S IER MASTER CLEAR TEST>
1193      ;*
1194      ;*      IER == "INTERRUPT ENABLE REGISTER"
1195      ;*
1196      ;* A 377 BYTE IS WRITTEN INTO IER AND THE REGISTER IS READ BACK AND CHECKED
1197      ;* FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND IER IS READ AND CHECKED FOR
1198      ;* 200.
1199      ;*****
1200

```

PROGRAM DOCUMENT

1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES A "MASTER CLEAR FAILURE" ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE DEVICE REGISTER CONTENTS :

CNDMA DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
MASTER CLEAR FAILURE

THE CONTENTS OF ALL BYTE SELECT REG'S ARE:

BSEL0	BSEL1	BSEL2	BSEL3
000	000	000	000
BSEL4	BSEL5	BSEL6	BSEL7
000	000	121	000
BSEL10	BSEL11	BSEL12	BSEL13
000	000	000	000
BSEL14	BSEL15	BSEL16	BSEL17
000	000	000	000

FOR OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE, AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

IF EXTENDED ERROR INFORMATION HAD BEEN INHIBITED USING THE IXE FLAG PRIOR TO RUNNING THE TEST, THE ABOVE ERROR WOULD HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :

CNDMA DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
MASTER CLEAR FAILURE

LISTING & ASSEMBLY CONTROL

```

1245          .SBTTL LISTING & ASSEMBLY CONTROL
1246
1247          000000      HELP=0          ; CONTROL LISTING OF HELP INFORMATION
1248                                     ; HELP=0   NO LIST
1249                                     ; HELP=1   LIST
1250
1256          002000      . =2000
1257
1258                                     .MCALL SVC
1259 002000      SVC                                     ; INITIALIZE SUPERVISOR MACROS
1260
1261 002000      BGNMOD  LU1MOD
1262
1263
1264          000001      $LSTIN= 1
1265          000001      $LSTTAG= 1
1266          000001      SVCINS= 1          ; LIST INSTRUCTIONS, SHIFTED RIGHT
1267          000001      SVCTST= 1         ; LIST TEST TAGS, SHIFTED RIGHT
1268          000001      SVCSUB= 1         ; LIST SUBTEST TAGS, SHIFTED RIGHT
1269          000001      SVCGBL= 1         ; LIST GLOBAL TAGS, SHIFTED RIGHT
1270          000001      SVCTAG= 1         ; LIST OTHER TAGS, SHIFTED RIGHT
1271
1272          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1273          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1274          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1275          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1276
1277 002000      POINTER BGNAU,BGNDU,ERRTBL
1278

```

PROGRAM HEADER

```

1287
1288
1289
1290
1291
1292
1293
1294
1295
1296 002000
      002000
      002000      103
      002001      116
      002002      104
      002003      115
      002004      101
      002005      000
      002006      000
      002007      000
      002010
      002010      101
      002011
      002011      060
      002012
      002012 000000
      002014
      002014 000156
      002016
      002016 040122
      002020
      002020 000000
      002022
      002022 002216
      002024
      002024 000000
      002026
      002026 040400
      002030
      002030 000000
      002032
      002032 000000
      002034
      002034 000000
      002036
      002036 000000
      002040
      002040 002124
      002042
      002042 000000
      002044
      002044 000000
      002046
      002046 000000
      002050
      002050      003
      002051      003
      002052

```

.SBTTL PROGRAM HEADER

!..

```

;THE PROGRAM HEADER MACRO CHARACTERIZES THIS DIAGNOSTIC. THE
;HEADER MACRO'S ARGUMENTS ARE FILE NAME, RELEASE LEVEL, PATCH
;DISPOSITION OF THE MOST RECENT PATCH, MAXIMUM TEST TIME IN SEC.,
;AND THE TYPE OF DIAGNOSTIC (0-SEQUENTIAL, 1 EXERCISER). THESE
;ARGUMENTS ARE IN RESPECTIVE ORDER.

```

!..

HEADER CNDMA,A,0,110.,0

```

L$NAME::
          .ASCII /C/
          .ASCII /N/
          .ASCII /D/
          .ASCII /M/
          .ASCII /A/
          .BYTE 0
          .BYTE 0
          .BYTE 0
L$REV::
          .ASCII /A/
L$DEPO::
          .ASCII /0/
L$UNIT::
          .WORD 0
L$TIML::
          .WORD 110.
L$MPCP::
          .WORD L$HARD
L$SPCP::
          .WORD 0
L$MPTP::
          .WORD L$HW
L$SPTP::
          .WORD 0
L$LADP::
          .WORD L$LAST
L$STA::
          .WORD 0
L$CO::
          .WORD 0
L$DTYP::
          .WORD 0
L$APT::
          .WORD 0
L$DTP::
          .WORD 0
L$PRIO::
          .WORD L$DISPATCH
L$ENVI::
          .WORD 0
L$EXP1::
          .WORD 0
L$MREV::
          .BYTE C$REVISION
          .BYTE C$EDIT
L$FF::

```

PROGRAM HEADER

002052 000000
 002054 000000
 002056
 002056 000000
 002060
 002060 003522
 002062
 002062 000000
 002064
 002064 000000
 002066
 002066 000000
 002070
 002070 020400
 002072
 002072 020374
 002074
 002074 000000
 002076
 002076 003542
 002100
 002100 104035
 002102
 002102 002236
 002104
 002104 017622
 002106
 002106 020356
 002110
 002110 020232
 002112
 002112 017614
 002114
 002114 000000
 002116
 002116 000000
 002120
 002120 000000

.WORD 0
 .WORD 0
 L\$SPC:: .WORD 0
 L\$DEVP:: .WORD L\$DVTYP
 L\$REPP:: .WORD 0
 L\$EXP4:: .WORD 0
 L\$EXP5:: .WORD 0
 L\$AUT:: .WORD 0
 L\$DUT:: .WORD L\$AU
 L\$LUN:: .WORD L\$DU
 L\$DESP:: .WORD 0
 L\$LOAD:: .WORD L\$DESC
 EMT E\$LOAD
 L\$ETP:: .WORD L\$ERRTBL
 L\$ICP:: .WORD L\$INIT
 L\$CCP:: .WORD L\$CLEAN
 L\$ACP:: .WORD L\$AUTO
 L\$PRT:: .WORD L\$PROT
 L\$TEST:: .WORD 0
 L\$DLY:: .WORD 0
 L\$HIME:: .WORD 0

1297
1303

.EVEN

DISPATCH TABLE

```

1305
1306
1307
1308
1309
1310
1311
1312 002122
      002122 000034
      002124
      002124 020402
      002126 021146
      002130 021256
      002132 021450
      002134 021612
      002136 021734
      002140 022136
      002142 022646
      002144 023214
      002146 024162
      002150 025142
      002152 025764
      002154 026046
      002156 026130
      002160 026232
      002162 027276
      002164 027360
      002166 027442
      002170 027526
      002172 027614
      002174 030102
      002176 030232
      002200 030362
      002202 030516
      002204 030646
      002206 030776
      002210 031134
      002212 036366
1313

```

.SBTTL DISPATCH TABLE

```

;////////////////////////////////////
;// THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
;// IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
;////////////////////////////////////

```

DISPATCH 28.

```

      .WORD 28
L$DISPATCH:
      .WORD T1
      .WORD T2
      .WORD T3
      .WORD T4
      .WORD T5
      .WORD T6
      .WORD T7
      .WORD T8
      .WORD T9
      .WORD T10
      .WORD T11
      .WORD T12
      .WORD T13
      .WORD T14
      .WORD T15
      .WORD T16
      .WORD T17
      .WORD T18
      .WORD T19
      .WORD T20
      .WORD T21
      .WORD T22
      .WORD T23
      .WORD T24
      .WORD T25
      .WORD T26
      .WORD T27
      .WORD T28

```

DEFAULT HARDWARE P TABLE

```

1321          .SBTTL DEFAULT HARDWARE P TABLE
1322
1323          ;////////////////////////////////////////////////////////////////////
1324          ;// THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1325          ;// THE TEST DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1326          ;// IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P TABLE.
1327          ;////////////////////////////////////////////////////////////////////
1328
1329          002214          BGNHW  DFPTBL
1330          002214          000007
1331          002216          L$HW::          .WORD  L10000 L$HW/2
1332          002216          DFPTBL::
1333
1334          002216          160020          .WORD  160020          ;DMV11 CSR UNIBUS ADDRESS
1335          002220          000300          .WORD  300          ;DMV11 INTERRUPT VECTOR
1336          002222          004000          .WORD  4000          ;DMV11 INTERRUPT PRIORITY LEVEL = 4
1337          002224          000000          .WORD  000          ;SWITCH REG. #1 (BOOT ADDRESS)
1338          002226          000000          .WORD  000          ;SWITCH REG. #2 (DDCMP ADDRESS)
1339          002230          000000          .WORD  0          ;H3254&H3255 USED
1340          002232          000111          .WORD  000111          ;MISC. CONTROLS:
1341
1342          ; POWER-UP MODE 0 MASK = 100
1343          ; 0 = NOT JUMPERED FOR MODE 0 POWER UP
1344          ; 1 = JUMPERED FOR MODE 0 POWER-UP <*** DEFAULT SETTING
1345          ; BOTH W5 & W6 REMOVED
1346
1347          ; BAUD RATE MASK = 77
1348          ; 7 = 19.2 K
1349          ; 11 = 56 K <*** DEFAULT SETTING
1350
1351          002234          ENDDHW
1352          002234
1353
1354          L10000:

```

SOFTWARE P TABLE

1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357 002234
 002234 000000
 002236
 002236
 1358 002236
 002236

.SBTTL SOFTWARE P TABLE

```

;////////////////////////////////////;////////////////////////////////////
;// THE SOFTWARE P TABLE CONTAINS THE VALUES OF THE PROGRAM
;// PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
;////////////////////////////////////;////////////////////////////////////

```

BGNSW SFPTBL

.WORD L10001 L\$SW/2

L\$SW::
 SFPTBL::

ENDSW

L10001:

GLOBAL EQUATES SECTION

1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368 002236

.SBTTL GLOBAL EQUATES SECTION

```

;////////////////////////////////////
;/ THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
;/ ARE USED IN MORE THAN ONE TEST.
;////////////////////////////////////
    
```

EQUALS

```

;
; BIT DEFINITIONS
;
100000 BIT15== 100000
040000 BIT14== 40000
020000 BIT13== 20000
010000 BIT12== 10000
004000 BIT11== 4000
002000 BIT10== 2000
001000 BIT09== 1000
000400 BIT08== 400
000200 BIT07== 200
000100 BIT06== 100
000040 BIT05== 40
000020 BIT04== 20
000010 BIT03== 10
000004 BIT02== 4
000002 BIT01== 2
000001 BIT00== 1

;
001000 BIT9== BIT09
000400 BIT8== BIT08
000200 BIT7== BIT07
000100 BIT6== BIT06
000040 BIT5== BIT05
000020 BIT4== BIT04
000010 BIT3== BIT03
000004 BIT2== BIT02
000002 BIT1== BIT01
000001 BIT0== BIT00

;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
000040 EF.START== 32. ; BIT POSITION IN SECOND STATUS WORD
000037 EF.RESTART== 31. ; (100000) START COMMAND WAS ISSUED
000036 EF.CONTINUE== 30. ; (040000) RESTART COMMAND WAS ISSUED
000035 EF.NEW== 29. ; (020000) CONTINUE COMMAND WAS ISSUED
000034 EF.PWR== 28. ; (010000) A NEW PASS HAS BEEN STARTED
; ; (004000) A POWER-FAIL/POWER-UP OCCURRED
;

;
; PRIORITY LEVEL DEFINITIONS
;
000340 PRI07== 340
000300 PRI06== 300
000240 PRI05== 240
000200 PRI04== 200
    
```

GLOBAL EQUATES SECTION

```

000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
;OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

1369
1370      .SBTTL  DEFINE THE NUMBER OF CSR'S
1371      C70020  CSREGS  = 16.
1372
1373
1374
1375      .SBTTL  NPR ADDRESS REGISTER EQUATES
1376      000070  NPRAOL  = 70          ;OUT NPR ADRS LO REG
1377      000071  NPRAOH  = NPRAOL+1    ;OUT NPR ADRS HI REG
1378      000072  NPRAOX  = NPRAOL+2    ;OUT NPR EXTENDED ADRS REG
1379      000074  NPRAIL  = NPRAOL+4    ;IN NPR ADRS LO REG
1380      000075  NPRAIH  = NPRAOL+5    ;IN NPR ADRS HI REG
1381      000076  NPRAIX  = NPRAOL+6    ;IN NPR EXTENDED ADRS REG
1382      000010  NPRBS7  = BIT3        ;"BANK SELECT 7" BIT   W/IN EXTENDED ADRS. REG.
1383
1384
1385
1386      .SBTTL  NPR DATA REG EQUATES
1387      123000  NPRDRL  = 123000      ;NPR DATA REGISTER - LOW BYTE
1388      123001  NPRDRH  = NPRDRL+1    ;NPR DATA REGISTER - HIGH BYTE
1389
1390
1391
1392      .SBTTL  NPR CONTROL REG EQUATES
1393      123004  NPRCTL  = NPRDRL+4    ;NPR CONTROL REGISTER
1394      000200  NPRABT  = BIT7        ;=1 IF BUS TIME-OUT ON NPR
1395      000100  NPRGO   = BIT6        ;SET FOR NOP, CLEAR TO "GO" / 0=DONE, 1=BUSY
1396      000040  NPRIO   = BIT5        ;0 = (LSI ==> DMV); 1 = (DMV ==> LSI)
1397      000020  LSIHLT  = BIT4        ;SETTING THIS WILL "HALT" THE LSI-11 !!
1398      000010  NPRBYT  = BIT3        ;SET TO 1 TO WRITE BYTE ONLY TO LSI 11
1399      000004  DMVPU   = BIT2        ;SET BY MICRO DIAG. MUST REMAIN SET!!!
1400      000002  LSIACL  = BIT1        ;IF SET, WILL CAUSE POWER DOWN CONDITION IN LSI!
1401      000001  DMVDAI  = BIT0        ;"DISABLE INIT" FROM EFFECTING DMV 11
1402
1403
1404      .SBTTL  NPR REQUEST FUNCTIONS

```

NPR REQUEST FUNCTIONS

```

1405      000004      NPRLD   = DMVPU           ;WORD XFER: LSI ==> DMV
1406      000044      NPRDL   = DMVPU!NPRI0      ;WORD XFER: DMV ==> LSI
1407      000054      NPRDLB  = DMVPU!NPRI0!NPRBYT ;BYTE XFER: DMV ==> LSI
1408
1409      ;
1410      ;
1411      .SBTTL  INTERRUPT REG EQUATES
1412      123005  IRQREG   = 123005      ;INTERRUPT REQUEST REG
1413      000004  IRQA    = BIT2        ;REQUEST BIT FOR XX0 INTERRUPT - 'A'
1414      000002  IRQB    = BIT1        ;REQUEST BIT FOR XX4 INTERRUPT  "B"
1415
1416      ;
1417      ;
1418      .SBTTL  CONTROL FLAGS FROM P TABLE ENTRIES
1419      000001  PU24    = BIT0        ;POWER-FAIL VECTURING MODE. 1 = MODE 0
1420                                     ; (I.E. JUMPERS W5 & W6 BOTH REMOVED)

```

SWITCH PACKS

1422
1423
1424
1425
1426
1427
1428
1429
1430

121000
121400

.SBTTL SWITCH PACKS

;;*****
;* SWITCH PACKS
;;*****

SWPBOT = 121000
SWPDDCMP = 121400

;"BOOT ADDRESS" SWITCH PACK [A200]
;"DDCMP ADDRESS" SWITCH PACK [A300]

CSR REG. DEFINITION FOR MAINT. LOOP

```

1432      .SBTTL  CSR REG. DEFINITION FOR MAINT. LOOP
1433
1434      ;*****
1435      .SBTTL      MAINTENANCE REGISTER - BSELO
1436      ;--*****
1437      ;          INTERRUPT ENABLE BITS
1438
1439      000001      IENBA   = BIT0           ;INTERRUPT ENABLE "A"
1440      000020      IENBB   = BIT4           ;INTERRUPT ENABLE "B"
1441
1442
1443      ;*****
1444      .SBTTL      MAINTENANCE REGISTER - BSEL1
1445      ; -*****
1446      ; MAINT. LOOP CONTROL BITS:
1447
1448      000200      RUN      = BIT7
1449      000100      MCLR    = BIT6
1450      000001      MREQ    = BIT0
1451
1452
1453      ;*****
1454      .SBTTL      MAINTENANCE REGISTER - BSEL2
1455      ;--*****
1456      ; MAINTENANCE FUNCTION CODES
1457
1458      000001      REDLOC   = 1           ;FUNCTION CODE FOR READ A 6502 LOCATION
1459      000002      WRILOC   = 2           ;FUNCTION CODE FOR WRITE A 6502 LOCATION
1460      000003      REDPAG   = 3           ;FUNCTION CODE FOR READ A 6502 MEMORY PAGE
1461      000004      WRIPAG   = 4           ;FUNCTION CODE FOR WRITE A 6502 RAM PAGE
1462      000005      EXECUT   = 5           ;FUNCTION CODE FOR EXECUTE AT GIVEN PC
1463
1464      000200      MRDY     = BIT7           ;M LOOP REDY FOR A COMMAND WHEN SET

```

DMV INTERNAL ADDRESSES

```

1466      .SBTTL  DMV INTERNAL ADDRESSES
1467
1468      ;+*****
1469      ;          DMV INTERNAL ADDRESSES
1470      ;--*****
1471
1472
1473      ;***** <<  MICROPROCESSOR REGISTER ADDRESS EQUATES  >> *****
1474
1475      .SBTTL          BYTE & WORD SELECT REGISTERS
1476
1477      000020      SLT0      =020
1478      000020      BSLT0     =SLT0
1479      000021      BSLT1     =SLT0+1
1480      000022      SLT2     =SLT0+2
1481      000022      BSLT2     =SLT0+2
1482      000023      BSLT3     =SLT0+3
1483      000024      SLT4     =SLT0+4
1484      000024      BSLT4     =SLT0+4
1485      000025      BSLT5     =SLT0+5
1486      000026      SLT6     =SLT0+6
1487      000026      BSLT6     =SLT0+6
1488      000027      BSLT7     =SLT0+7
1489
1490      .SBTTL          VIA'S REGISTERS
1491
1492      120000      ORB       =120000
1493      120001      ORA       =ORB+1
1494      120002      DDRB      =ORB+2
1495      120003      DDRA      =ORB+3
1496      120004      T1CL      =ORB+4
1497      120005      T1CH      =ORB+5
1498      120005      T1LHGO     =ORB+5
1499      120006      T1LL      =ORB+6
1500      120007      T1LH      =ORB+7
1501      120010      T2LL      =ORB+10
1502      120010      T2CL      =T2LL
1503      120011      T2CH      =ORB+11
1504      120012      SR        =ORB+12
1505      120013      ACR       =ORB+13
1506      120014      PCR       =ORB+14
1507      120015      IFR       =ORB+15
1508      120016      IENR      =ORB+16
1509      120017      ORAM      =ORB+17
1510
1511      .SBTTL          VIA'S "IFR" REGISTER'S BIT ASSIGNMENTS
1512
1513      000200      IFRIRQ    =BIT7  ;"IRQ" HAS BEEN ISSUED -- LOGICAL "OR" OF BITS 0  > 6
1514      000100      IFRT1     =BIT6  ;"T1" -- TIMER # 1 TIMED-OUT
1515      000040      IFRT2     =BIT5  ;"T2" -- TIMER # 1 TIMED-OUT
1516      000020      IFRCB1    =BIT4  ;"CB1" EDGE DETECTED ("K2 LINE UNIT STEP" O/P SIGNAL FROM SR)
1517      000010      IFRCB2    =BIT3  ;"CB2" EDGE DETECTED (UNUSED!)
1518      000004      IFRSR     =BIT2  ;"SR" REGISTER COMPLETED SHIFT OPERATION
1519      000002      IFRCA1    =BIT1  ;"CA1" EDGE DETECTED ("K6 MOD RDY H")
1520      000001      IFRCA2    =BIT0  ;"CA2" EDGE DETECTED ("K2 CTS H")
1521
1522

```

GLOBAL DATA SECTION

1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780
 1781
 1782
 1783
 1784 002236
 002236
 002236 000000
 002240 000000
 002242 000000
 002244 000000
 1785
 1786
 1787
 1788
 1789 002246
 1790 002246 000000
 1791 002250
 1792 002250 000000
 1793 002252
 1794 002252 000000
 1795 002254
 1796 002254 000000
 1797 002256
 1798 002256 000000
 1799 002260
 1800 002260 000000
 1801 002262
 1802 002262 000000
 1803 002264
 1804 002264 000000
 1805 002266 000000
 1806 002270 000000
 1807 002272 000000
 1808 002274 000000
 1809 002276 000000
 1810 002300 000000
 1811 002302 000000
 1812 002304 000000
 1813
 1814
 1815
 1816
 1817 002306 000000
 1818 002310 000000
 1819 002312 000000
 1820 002314 000000
 1821 002316 110400
 1822 002320 000007
 1823 002322 000000
 1824 002324 000000

```

.SBTTL GLOBAL DATA SECTION
;/////////////////////////////////////////////////////////////////
; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
; IN MORE THAN ONE TEST.
;/////////////////////////////////////////////////////////////////
;*****
.SBTTL CONTROL BLOCK FOR STACKED ERROR MESSAGES
;-----
                ERRRTBL
                                L$ERRRTBL::
ERRRTYP::      .WORD  0
ERRNBR::      .WORD  0
ERRMSG::      .WORD  0
ERRBLK::      .WORD  0
;*****
.SBTTL STORAGE FOR DEVICE REGISTERS
;-----
WSR0:
BSR0:  .WORD  0
WSR2:
BSR1:  .WORD  0
WSR4:
BSR2:  .WORD  0
WSR6:
BSR3:  .WORD  0
WSR10:
BSR4:  .WORD  0
WSR12:
BSR5:  .WORD  0
WSR14:
BSR6:  .WORD  0
WSR16:
BSR7:  .WORD  0
BSR10: .WORD  0
BSR11: .WORD  0
BSR12: .WORD  0
BSR13: .WORD  0
BSR14: .WORD  0
BSR15: .WORD  0
BSR16: .WORD  0
BSR17: .WORD  0
;*****
.SBTTL MISCELLANEOUS STORAGE
;-----
TDATA: .WORD  0      ;TEST DATA
GDATA: .WORD  0      ;EXPECTED DATA
BDATA: .WORD  0      ;ACTUAL DATA
XDATA: .WORD  0      ;EXCLUSIVE OR BETWEEN "GDATA" & "BDATA"
DELAY1: .WORD 110400  ;DELAY TIME, 3 INST., 500 MILLISEC.
DELAY2: .WORD  7      ;DELAY TIME FOR M-LOOP FUNCTION, 100 USEC.APPROX.
LOGDEV: .WORD  0      ;LOGICAL DEVICE NUMBER
PSTACK: .WORD  0      ;CONTAINS BASE LEVEL PROGRAM STACK POINTER

```

MISCELLANEOUS STORAGE

```

1825 002326 000000      INTFLG: .WORD 0      ;INTERRUPT RECEIVED FLAG BYTES. ALLOCATION:
1826                                     ; LOW BYTE FOR "A" & HIGH BYTE FOR "B"
1827 002330 000000      INTWCH: .WORD 0      ;BYTE IS SET NON-ZERO WHEN HANDLER SHOULD BE
1828                                     ; WATCHING FOR INT'S. ALLOCATION: SEE INTFLG
1829 002332 000000      ERRFLG: .WORD 0      ;ERROR FLAG
1830 002334 000000      REGNUM: .WORD 0      ;REGISTER NUMBER -- FOR PASSING ARG. TO "ERR#"
1831 002336 000000      FRSTIM: .WORD 0      ;FLAG=0 IF PROGRAM JUST LOADED
1832 002340 000000      FRSPAS: .WORD 0      ;FLAG=0 IF FIRST PASS AFTER LOAD
1833 002342 000000      DEVMAP: .WORD 0      ;BIT MAP OF ACTIVE DEVICES
1834 002344 000000      DEVPTR: .WORD 0      ;DEVICE MAP BIT POINTER
1835 002346 000000      CONSOL: .WORD 0      ;CONSOLE DEVICE FLAG -- NON ZERO = NONE PRESENT
1836 002350 000000      PFLAG: .WORD 0      ;MISC. PROGRAM FLAGS
1837                                     ;
1838                                     ; THE ABOVE WORD CONTAINS MISC. FLAGS WHICH CAN ONLY BE ACCESSED BY PATCHING.
1839                                     ; IT IS NOT INTENDED THAT THEY BE SET OR CLEARED EXCEPT UNDER VERY UNUSUAL
1840                                     ; CIRCUMSTANCES. THEREFORE, THEY WILL NOT BE DOCUMENTED ANY OTHER PLACE
1841                                     ; EXCEPT RIGHT HERE.
1842                                     ;
1843                                     ; BIT 0 -- WHEN SET, THOSE TESTS WHICH DO A BUS RESET WILL NOT BE EXECUTED.
1844                                     ; THIS WAS IMPLEMENTED TO SAVE WEAR & TEAR ON THE RX01 IN THE
1845                                     ; DEVELOPMENT SYSTEM WHILE DOING LONG TERM TESTING OF ALL OTHER
1846                                     ; TESTS.
1847                                     ;
1848                                     ; BIT 1 -- CPU TYPE. (NOT USED)
1849                                     ;
1850                                     ; BIT 2 -- CONTROLS PRINTING OF EXTENDED ERROR INFORMATION DURING "MOVING
1851                                     ; INVERSIONS TEST" OF RAM. NORMALLY ONLY ADDRESS, GOOD & BAD
1852                                     ; DATA, AND XOR WILL BE PRINTED. IF THIS BIT IS SET HOWEVER,
1853                                     ; INFORMATION IDENTIFYING WHERE WITHIN THE ALGORITHM THE ERROR
1854                                     ; WAS DETECTED IS REPORTED. THE FOLLOWING ABBREVIATIONS ARE USED
1855                                     ; IN THE HEADING:
1856                                     ; BIT --- IDENTIFIES THE INNERMOST LOOP. WHICH BIT IS
1857                                     ; BEING INVERTED AT EACH LOCATION. BITS ARE
1858                                     ; IDENTIFIED AS 0 THROUGH 7.
1859                                     ; DATA -- IDENTIFIES THE VALUE TO WHICH THE ABOVE BIT IS
1860                                     ; BEING SET (I.E. 0 OR 1). IT IS FIRST READ AND
1861                                     ; CHECKED FOR EXPECTED CONTENTS; THEN THE BIT IS
1862                                     ; INVERTED TO THIS STATE (DATA) AND RE-WRITTEN;
1863                                     ; THEN IT IS AGAIN READ & CHECKED FOR THE NEW
1864                                     ; VALUE.
1865                                     ; SEQ --- INDICATES THE DIRECTION (FWD OR BKWD) THE TEST
1866                                     ; WAS SCANNING THROUGH RAM WHEN THE ERROR OCCURED.
1867                                     ; LSB --- THIS IS THE LOGICAL LEAST SIGNIFICANT BIT OF THE
1868                                     ; RAM ADDRESS AS WE SCAN THROUGH MEMORY. BY
1869                                     ; VARYING THIS, THE ALGORITHM GENERATS NON-SEQUEN-
1870                                     ; TIAL ADDRESSING OF RAM AND EFFECTS A MUCH MORE
1871                                     ; THOROUGH TEST OF MEMORY.
1872                                     ;
1873                                     ;
1874                                     ;

```


CURRENT DEVICE PARAMETERS

```

1876          .SBTTL CURRENT DEVICE PARAMETERS
1877
1878          160000          $MPCSR  ..          160000          ;INITIAL ASSEMBLED IN CSR ADDRESS
1879
1880 002352          MPCSR:          ;POINTER TO THE DMV11 CSR'S
1881 002352          BSEL0:          ;POINTER TO BSEL0
1882 002352          BSEL:          ;ALTERNATE NAME FOR BSEL0
1883 002352 160000          SEL0: .WORD $MPCSR          ;POINTER TO SEL0
1884 002354 160001          BSEL1: .WORD $MPCSR.1          ;POINTER TO BSEL1
1885 002356          BSEL2:          ;POINTER TO BSEL2
1886 002356 160002          SEL2: .WORD $MPCSR.2          ;POINTER TO SEL2
1887 002360 160003          BSEL3: .WORD $MPCSR.3          ;POINTER TO BSEL3
1888 002362          BSEL4:          ;POINTER TO BSEL4
1889 002362 160004          SEL4: .WORD $MPCSR.4          ;POINTER TO SEL4
1890 002364 160005          BSEL5: .WORD $MPCSR.5          ;POINTER TO BSEL5
1891 002366          BSEL6:          ;POINTER TO BSEL6
1892 002366 160006          SEL6: .WORD $MPCSR.6          ;POINTER TO SEL6
1893 002370 160007          BSEL7: .WORD $MPCSR.7          ;POINTER TO BSEL7
1894 002372          BSEL10:          ;POINTER TO BSEL10
1895 002372 160010          SEL10: .WORD $MPCSR.10          ;POINTER TO SEL10
1896 002374 160011          BSEL11: .WORD $MPCSR.11          ;POINTER TO BSEL11
1897 002376          BSEL12:          ;POINTER TO BSEL12
1898 002376 160012          SEL12: .WORD $MPCSR.12          ;POINTER TO SEL12
1899 002400 160013          BSEL13: .WORD $MPCSR.13          ;POINTER TO BSEL13
1900 002402          BSEL14:          ;POINTER TO BSEL14
1901 002402 160014          SEL14: .WORD $MPCSR.14          ;POINTER TO SEL14
1902 002404 160015          BSEL15: .WORD $MPCSR.15          ;POINTER TO BSEL15
1903 002406          BSEL16:          ;POINTER TO BSEL16
1904 002406 160016          SEL16: .WORD $MPCSR.16          ;POINTER TO SEL16
1905 002410 160017          BSEL17: .WORD $MPCSR.17          ;POINTER TO BSEL17
1906
1907 002412 000300          MPIVEC: .WORD 300          ;DMV11 INPUT INTERRUPT VECTOR
1908 002414 000304          MPOVEC: .WORD 304          ;DMV11 OUTPUT INTERRUPT VECTOR
1909 002416 000340          MPRIOR: .WORD 340          ;DMV11 DEVICE PRIORITY
1910
1911          .SBTTL GEN'L PURPOSE SCRATCH STORAGE
1912
1913 002420 000000          REG0: .WORD 0
1914 002422 000000          REG1: .WORD 0
1915 002424 000000          REG2: .WORD 0
1916 002426 000000          REG3: .WORD 0
1917 002430 000000          REG4: .WORD 0
1918 002432 000000          REG5: .WORD 0
1919 002434 000000          REG6: .WORD 0
1920 002436 000000          REG7: .WORD 0
1921

```

SCRATCH STORAGE FOR MESSAGE REPORTING.

1923
 1924
 1925 002440 000000
 1926 002442 000000
 1927 002444 000000
 1928 002446 000000
 1929 002450 000000
 1930 002452 000000
 1931 002454 000000
 1932 002456 000000
 1933 002460 000000
 1934 002462 000000
 1935 002464 000000
 1936 002466 000000
 1937 002470 000000
 1938 002472 000000
 1939 002474 000000
 1940 002476 000000
 1941 002500 000000
 1942 002502 000000
 1943
 1944

.SBTTL SCRATCH STORAGE FOR MESSAGE REPORTING.

TMP0: .WORD 0
 TMP1: .WORD 0
 TMP2: .WORD 0
 TMP3: .WORD 0
 TMP4: .WORD 0
 TMP5: .WORD 0
 TMP6: .WORD 0
 TMP7: .WORD 0
 TMP8: .WORD 0
 TMP9: .WORD 0
 TMPA: .WORD 0
 TMPB: .WORD 0
 TMPC: .WORD 0
 TMPD: .WORD 0
 TMPE: .WORD 0
 TMPF: .WORD 0
 FWPC: .WORD 0
 OLDSP: .WORD 0

;SAVE LOCATION FOR A "PC" VALUE RESET
 ;SAVE LOCATION FOR A STACK POINTER RESET VALUE

***** DATA PATTERN A *****

1946
 1947
 1948
 1949 002504 000020
 1950 002506 001
 1951 002507 002
 1952 002510 004
 1953 002511 010
 1954 002512 020
 1955 002513 040
 1956 002514 100
 1957 002515 200
 1958 002516 052
 1959 002517 300
 1960 002520 140
 1961 002521 060
 1962 002522 030
 1963 002523 014
 1964 002524 006
 1965 002525 003
 1966
 1967
 1968
 1969
 1970 002526 000026
 1971 002530 125
 1972 002531 252
 1973 002532 000
 1974 002533 377
 1975 002534 001
 1976 002535 002
 1977 002536 004
 1978 002537 010
 1979 002540 020
 1980 002541 040
 1981 002542 100
 1982 002543 200
 1983 002544 376
 1984 002545 375
 1985 002546 373
 1986 002547 367
 1987 002550 357
 1988 002551 337
 1989 002552 277
 1990 002553 177
 1991 002554 000
 1992

.SBTTL ***** DATA PATTERN A *****

.EVEN				.USAGE:
PATA:	.WORD	PATB PATA 2		;# OF BYTES IN PATTERN
	.BYTE	001		;BSEL0
	.BYTE	002		;BSEL1
	.BYTE	004		;BSEL2
	.BYTE	010		;BSEL3
	.BYTE	020		;BSEL4
	.BYTE	040		;BSEL5
	.BYTE	100		;BSEL6
	.BYTE	200		;BSEL7
	.BYTE	052		;BSEL10
	.BYTE	300		;BSEL11
	.BYTE	140		;BSEL12
	.BYTE	060		;BSEL13
	.BYTE	030		;BSEL14
	.BYTE	014		;BSEL15
	.BYTE	006		;BSEL16
	.BYTE	003		;BSEL17

.SBTTL ***** DATA PATTERN B *****

.EVEN				.USAGE:
PATB:	.WORD	PATC-PATB-2		;# OF BYTES IN PATTERN
	.BYTE	125		
	.BYTE	252		
	.BYTE	000		
	.BYTE	377		
	.BYTE	001		
	.BYTE	002		
	.BYTE	004		
	.BYTE	010		
	.BYTE	020		
	.BYTE	040		
	.BYTE	100		
	.BYTE	200		
	.BYTE	376		
	.BYTE	375		
	.BYTE	373		
	.BYTE	367		
	.BYTE	357		
	.BYTE	337		
	.BYTE	277		
	.BYTE	177		
	.BYTE	000		

***** DATA PATTERN C *****

```

1994
1995
1996
1997
1998
1999 002556 000012
2000 002560 002 377
2001 002562 003 366
2002 002564 000 100
2003 002566 013 040
2004 002570 006 106
2005 002572 007 107
2006 002574 012 112
2007 002576 014 042
2008 002600 015 000
2009 002602 016 200
2010
2011
2012
2013 002604 100
2014 002605 000
2015 002606 377
2016 002607 56
2017 002610 000
2018 002611 000
2019 002612 106
2020 002613 107
2021 002614 000
2022 002615 000
2023 002616 112
2024 002617 040
2025 002620 042
2026 002621 000
2027 002622 200
2028 002623 000
2029
2030
2031
2032
2033
2034 002624 000
2035 002625 377
2036 002626 000
2037 002627 000
2038 002630 377
2039 002631 377
2040 002632 000
2041 002633 000
2042 002634 377
2043 002635 377
2044 002636 000
2045 002637 000
2046 002640 000
2047 002641 377
2048 002642 200
2049 002643 377

```

```

.SBTTL ***** DATA PATTERN C *****
; USED BY TEST # 11 TO LOAD UP THE VIA'S REGISTERS. THE REGISTER NUMBER
; LOADED IS THE FIRST BYTE AND THE VALUE LOADED INTO IT IS THE SECOND BYTE

.EVEN
PATC: .WORD <PATCR-PATC-2>/2
      .BYTE 2,377 ;SETUP ORB AS AN I/O (READ/WRITE) REGISTER
      .BYTE 3,366 ;SETUP ORA AS AN O/P REGISTER - IT CAN'T BE TESTED!
      .BYTE 0,100 ;LOAD UP ORB
      .BYTE 13,040 ; ACR
      .BYTE 6,106 ; T1LL
      .BYTE 7,107 ; T1LH
      .BYTE 12,112 ; SR
      .BYTE 14,042 ; PCR
      .BYTE 15,000 ; IFR
      .BYTE 16,200 ; IER

; THIS TABLE IS THE LIST OF EXPECTED CONTENTS OF THE VIA'S REGISTERS
PATCR: .BYTE 100 ; ORB
      .BYTE 000 ; ORA
      .BYTE 377 ; DDRB
      .BYTE 366 ; DDRA
      .BYTE 000 ; T1CL
      .BYTE 000 ; T1CH
      .BYTE 106 ; T1LL
      .BYTE 107 ; T1LH
      .BYTE 000 ; T2CL
      .BYTE 000 ; T2CH
      .BYTE 112 ; SR
      .BYTE 040 ; ACR
      .BYTE 042 ; PCR
      .BYTE 000 ; IFR
      .BYTE 200 ; IER
      .BYTE 000 ; ORA

; THIS IS THE TABLE OF TEST PATTERN "A" MASKS. BEFORE A REGISTER'S
; CONTENTS IS TESTED, A BICB IS DONE USING ITS RESPECTIVE BYTE FROM
; THE TABLE BELOW (INSURING THAT "DON'T CARE" BITS ARE IGNORED).
PATCM: .BYTE 000 ; ORB
      .BYTE 377 ; ORA -- THIS REGISTER CAN'T BE TESTED!!!
      .BYTE 000 ; DDRB
      .BYTE 000 ; DDRA
      .BYTE 377 ; T1CL -- THIS IS A FREE RUNNING COUNTER
      .BYTE 377 ; T1CH -- THIS IS A FREE RUNNING COUNTER
      .BYTE 000 ; T1LL
      .BYTE 000 ; T1LH
      .BYTE 377 ; T2CL -- THIS IS A FREE RUNNING COUNTER
      .BYTE 377 ; T2CH -- THIS IS A FREE RUNNING COUNTER
      .BYTE 000 ; SR
      .BYTE 000 ; ACR
      .BYTE 000 ; PCR
      .BYTE 377 ; IFR
      .BYTE 200 ; IER - BIT 7 IS ALWAYS READ AS ZERO
      .BYTE 377 ; ORA - THIS REGISTER CAN'T BE TESTED!!!

```

***** DATA PATTERN D *****

2051			.SBTTL ***** DATA PATTERN D *****
2052			
2053			.EVEN
2054	002644	000026	PATD: .WORD PATE-PATD 2
2055	002646	200	.BYTE 200
2056	002647	201	.BYTE 201
2057	002650	202	.BYTE 202
2058	002651	204	.BYTE 204
2059	002652	210	.BYTE 210
2060	002653	220	.BYTE 220
2061	002654	240	.BYTE 240
2062	002655	300	.BYTE 300
2063	002656	200	.BYTE 200
2064	002657	000	.BYTE 000
2065	002660	001	.BYTE 001
2066	002661	002	.BYTE 002
2067	002662	004	.BYTE 004
2068	002663	010	.BYTE 010
2069	002664	020	.BYTE 020
2070	002665	040	.BYTE 040
2071	002666	100	.BYTE 100
2072	002667	000	.BYTE 000
2073	002670	325	.BYTE 325
2074	002671	125	.BYTE 125
2075	002672	252	.BYTE 252
2076	002673	052	.BYTE 052

2077			
2078			
2079			.SBTTL ***** DATA PATTERN E *****
2080			
2081			.EVEN
2082	002674	000026	PATE: .WORD PATF-PATE-2
2083	002676	200	.BYTE 200
2084	002677	201	.BYTE 201
2085	002700	203	.BYTE 203
2086	002701	207	.BYTE 207
2087	002702	217	.BYTE 217
2088	002703	237	.BYTE 237
2089	002704	277	.BYTE 277
2090	002705	377	.BYTE 377
2091	002706	377	.BYTE 377
2092	002707	377	.BYTE 377
2093	002710	376	.BYTE 376
2094	002711	374	.BYTE 374
2095	002712	370	.BYTE 370
2096	002713	360	.BYTE 360
2097	002714	340	.BYTE 340
2098	002715	300	.BYTE 300
2099	002716	200	.BYTE 200
2100	002717	200	.BYTE 200
2101	002720	325	.BYTE 325
2102	002721	200	.BYTE 200
2103	002722	252	.BYTE 252
2104	002723	200	.BYTE 200

***** DATA PATTERN F *****

2106			.SBTTL	***** DATA PATTERN F *****
2107				
2108			.EVEN	
2109	002724	000045	PATF:	.WORD <PATG-PATF-2>/2
2110	002726	125252		.WORD 125252
2111	002730	052525		.WORD 052525
2112	002732	000000		.WORD 000000
2113	002734	177777		.WORD 177777
2114	002736	000001		.WORD 000001
2115	002740	000002		.WORD 000002
2116	002742	000004		.WORD 000004
2117	002744	000010		.WORD 000010
2118	002746	000020		.WORD 000020
2119	002750	000040		.WORD 000040
2120	002752	000100		.WORD 000100
2121	002754	000200		.WORD 000200
2122	002756	000400		.WORD 000400
2123	002760	001000		.WORD 001000
2124	002762	002000		.WORD 002000
2125	002764	004000		.WORD 004000
2126	002766	010000		.WORD 010000
2127	002770	020000		.WORD 020000
2128	002772	040000		.WORD 040000
2129	002774	100000		.WORD 100000
2130	002776	177776		.WORD 177776
2131	003000	177775		.WORD 177775
2132	003002	177773		.WORD 177773
2133	003004	177767		.WORD 177767
2134	003006	177757		.WORD 177757
2135	003010	177737		.WORD 177737
2136	003012	177677		.WORD 177677
2137	003014	177577		.WORD 177577
2138	003016	177377		.WORD 177377
2139	003020	176777		.WORD 176777
2140	003022	175777		.WORD 175777
2141	003024	173777		.WORD 173777
2142	003026	167777		.WORD 167777
2143	003030	157777		.WORD 157777
2144	003032	137777		.WORD 137777
2145	003034	077777		.WORD 077777
2146	003036	000000		.WORD 000000

***** DATA PATTERN F *****

2148 003040
 2149
 2150
 2151
 2152
 2153 003040 000020
 2154 003042 000
 2155 003043 200
 2156 003044 000
 2157 003045 000
 2158 003046 033
 2159 003047 000
 2160 003050 305
 2161 003051 000
 2162 003052 000
 2163 003053 000
 2164 003054 000
 2165 003055 000
 2166 003056 000
 2167 003057 000
 2168 003060 000
 2169 003061 000
 2170
 2171
 2172
 2173 003062
 2174

PATG:

.SBTTL ***** DATA PATTERN RESULTS TABLE FOR MASTER CLEAR (RESFMC) *****

.EVEN			
RESFMC:	.WORD	RESFT3-RESFMC-2	
BSELRS:	.BYTE	000	;BSEL0
	.BYTE	200	;BSEL1 -- "RUN" BIT SET
	.BYTE	000	;BSEL2
	.BYTE	000	;BSEL3
	.BYTE	033	;BSEL4 -- CODE FOR THE DMV 11
	.BYTE	000	;BSEL5
	.BYTE	305	;BSEL6 -- INDICATING VALID COMPLETION OF U-DIAG.
	.BYTE	000	;BSEL7
	.BYTE	000	;BSEL10
	.BYTE	000	;BSEL11
	.BYTE	000	;BSEL12
	.BYTE	000	;BSEL13
	.BYTE	000	;BSEL14
	.BYTE	000	;BSEL15
	.BYTE	000	;BSEL16
	.BYTE	000	;BSEL17

.SBTTL ***** DATA PATTERN RESULTS FOR TEST 3 (RESFT3) *****

RESFT3: .BLKW 16.
 .EVEN

DATA BUFFER AREAS

```
2176          .SBTTL DATA BUFFER AREAS
2177
2178 003122     BUFAREA:          .BLKB 256.
2179
2180
2181           ; THIS BUFFER HAS SOME ALTERNATE USES TOO. THE FOLLOWING LABELS ARE PROVIDED
2182           ; FOR THOSE USAGES.
2183
2184           003322     W0 = BUFAREA+128.          ;THIS WORD TABLE STARTS IN THE MIDDLE OF "BUFAREA"
2185           003324     W1 = W0+2                ;AND IS USED BY "ERR6" FOR PRINTING BYTES
2186           003326     W2 = W1+2
2187           003330     W3 = W2+2
2188           003332     W4 = W3+2
2189           003334     W5 = W4+2
2190           003336     W6 = W5+2
2191           003340     W7 = W6+2
2192           003342     W8 = W7+2
2193           003344     W9 = W8+2
2194           003346     WA = W9+2
2195           003350     WB = WA+2
2196           003352     WC = WB+2
2197           003354     WD = WC+2
2198           003356     WE = WD+2
2199           003360     WF = WE+2
2200
2201           003122     BT1 = BUFAREA              ;BYTE TABLE # 1
2202           003206     BT2 = BUFAREA+64          ;BYTE TABLE # 2
```


GLOBAL TEXT SECTION

.SBTTL GLOBAL TEXT SECTION

```

2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215

```

; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
; MORE THAN ONE TEST.

```

; *****
; * NAMES OF DEVICES SUPPORTED BY PROGRAM
; - - - - -
DEVTYP <M8053 OR M8064>

```

```

L$DVTYP::
.ASCIZ #M8053 OR M8064#

```

```

003522
003522      115      070      060
003525      065      063      040
003530      117      122      040
003533      115      070      060
003536      066      064      000

```

.EVEN

```

2216
2217
2218
2219
2220
2221
2222

```

; * TITLE OF PROGRAM
; - - - - -

```

.RADIX 10.
DESCRIPT <DMV-11 U-CONTRL LOGIC DIAG - PART 1 OF 2>

```

```

L$DESC::
.ASCIZ /DMV-11 U CONTRL LOG

```

```

000012
003542
003542
003542      104      115      126
IC DIAG - PART 1 OF 2/
003545      055      061      061
003550      040      125      055
003553      103      117      116
003556      124      122      114
003561      040      114      117
003564      107      111      103
003567      040      104      111
003572      101      107      040
003575      055      040      120
003600      101      122      124
003603      040      061      040
003606      117      106      040
003611      062      000

```

.EVEN

```

000010
.RADIX 8.
2223
2224

```

GLOBAL SUBROUTINES

```

2232 .SBTTL GLOBAL SUBROUTINES
2233
2234 ;////////////////////////////////////
2235 ;/ THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
2236 ;////////////////////////////////////
2237
2238 ;*****
2239 .SBTTL MASCLR MASTER CLEAR SUBROUTINE
2240
2241 ;
2242 ; FUNCTION:
2243 ;
2244 ; THIS SUBROUTINE FORCES THE 6502 MICROPROCESSOR TO EXECUTE A MINI 17 PART
2245 ; DIAGNOSTIC OF THE MICRO-PROCESSOR INSTRUCTION SET, RAM DATA AND ADDRESSING
2246 ; VALIDITY, AND A ROM CRC TEST. THE CLEAR SUBROUTINE EXECUTES IN
2247 ; APPROXIMATELY 500 HUNDRED(S) MILLISECOND. THIS SUBROUTINE WILL SEND THE
2248 ; MASTER CLEAR COMMAND AND DELAY FOR APPROX. 500 MSEC. AT WHICH POINT IN
2249 ; TIME, THE STATE OF THE CSR REGISTERS IS TESTED. IF ANY ONE OF THE
2250 ; REGISTERS CONTAINS ANYTHING THAT IS NOT EXPECTED, AN ERROR IS QUEUE UP AND
2251 ; THE CARRY BIT IS SET. ELSE, THE CARRY BIT IS CLEARED.
2252
2253 ; CALLING SEQUENCE:
2254 ;
2255 ; JSR PC,MASCLR
2256 ; BCC N$ ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2257 ; ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
2258 ; <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
2259 ; N$: <RESUMPTION OF NORMAL PROCESSING>
2260 ;
2261 ; - *****
2262 ;
2263 003614 010146 MASCLR: MOV R1,-(SP) ; SAVE REGISTER ONE
2264
2265 003616 112777 000300 176530 MOVB #RUN!MCLR,@BSEL1 ;SET BOTH THE RUN AND MASTER CLEAR BITS
2266 ;TO INITIATE THE MICRODIAGNOSTIC
2267
2268 ;NOW DELAY LONG ENOUGH FOR THE MICRODIAGNOSTIC TO COMPLETE
2269
2270 003624 013701 002316 MOV DELAY1,R1 ;INITIALIZE THE LOOP COUNTER FOR DELAY LOOP
2271 003630 001402 2$: BEQ 1$ ; EXIT DELAY LOOP IF THE TIME HAS EXPIRED
2272 003632 005301 DEC R1 ; ELSE, DECREMENT THE LOOP COUNTER AND
2273 003634 000775 BR 2$ ; CONTINUE TO LOOP.
2274 003636 1$: ; TIME-UP!
2275 003636 132777 000200 176510 BITB #RUN,@BSEL1 ;CHECK THE RUN BIT
2276 003644 001410 BEQ 3$ ;IF NOT SET, GO REPORT THE ERROR
2277
2278 ;IF THE RUN BIT IS SET, MICRODIAGNOSTICS ARE COMPLETE.
2279 ;CHECK IF ALL MICRODIAGNOSTICS PASSED.
2280
2281 003646 127737 176514 003050 4$: CMPB @BSEL6,BSELRS+6 ;THIS CHECKS THE BYTE IN B-SELECT 6 FOR THE
2282 ;VALID MICRODIAGNOSTIC COMPLETION CODE.
2283 003654 001004 BNE 3$ ;IF BAD, GO REPORT ERROR
2284
2285 003656 127737 176500 003046 CMPB @BSEL4,BSELRS+4 ;ELSE, CHECK FOR THE VALID CODE FOR A DMV 11
2286 003664 001420 BEQ 6$ ;IF THIS TOO IS CORRECT THEN NO ERROR EXISTS
2287 ;ELSE, FALL INTO THE ERROR REPORTING CODE
2288

```

MASCLR MASTER CLEAR SUBROUTINE

```

2289 003666 004737 004434      3$: JSR PC,GETBSR      ;GET THE BSEL REGISTERS FOR DUMPING
2290 003672                                GTDF 20$,ERR3      ;MASTER CLEAR ERROR
                                ;      QUEUE "DEVICE FATAL" ERROR # 1
                                MOV #T.EDF,ERRTYP
                                MOV #1,ERRNBR
                                MOV #20$,ERRMSG
                                MOV #ERR3,ERRBLK
003672 012737 000001 002236
003700 012737 000001 002240
003706 012737 003734 002242
003714 012737 005414 002244
2291 003722 000261          SEC          ;INDICATE TO THE CALLING ROUTINE THAT
2292 003724 000401          BR 7$      ; AN ERROR WAS DETECTED
2293
2294 003726 000241          6$: CLC          ;CLEAR THE CARRY BIT TO INDICATE NO ERROR
2295 003730 012601          7$: MOV (SP)+,R1      ;RESTORE REGISTER ONE
2296 003732 000207          RTS PC      ; RETURN TO THE CALLER
2297          .NLIST      BEX
2298 003734 115 101 123 20$: .ASCIZ /MASTER CLEAR FAILURE/
2299          .LIST      BEX
2300          .EVEN

```

M-LOOP MSTCLR MASTER CLEAR & ENTER M-LOOP

```

2302 .SBTTL M-LOOP -- MSTCLR -- MASTER CLEAR & ENTER M-LOOP
2303 ;*****
2304 ; MSTCLR -- MASTER CLEAR & ENTER M-LOOP
2305 ;
2306 ; CALLING SEQUENCE:
2307 ;
2308 ; JSR PC,MSTCLR
2309 ; BCC N$ ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2310 ; ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
2311 ; <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
2312 ;
2313 ; N$: <RESUMPTION OF NORMAL PROCESSING>
2314 ;
2315 ;-----*****
2316
2317 003762 012777 140400 176362 MSTCLR: MOV #<RUN!MCLR!MREQ>*256.,@SELO ;INITIATE M-LOOP
2318
2319 003770 010346 MOV R3,-(SP)
2320 003772 012703 000014 MOV #12.,R3 ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2321 003776 077301 1$: SOB R3,1$
2322 004000 012603 MOV (SP)+,R3
2323
2324 004002 132777 000200 176346 BITB #MRDY,@BSEL2 ;DID THE M-LOOP FINISH
2325 004010 001023 BNE 5$ ;YES, GOOD. RETURN
2326 004012 004737 004576 JSR PC,GETWSR ;GET BYTE SELECT REGISTERS
2327 004016 012737 000301 002310 MOV #RUN!MCLR!MREQ,GDATA ;IDENTIFY REQUESTED FUNCTION
2328 004024 GTDF EM3,ERR4 ;"MRDY" TIMEOUT
; QUEUE "DEVICE FATAL" ERROR # 2
004024 012737 000001 002236 MOV #T.EDF,ERRTYP
004032 012737 000002 002240 MOV #2,ERRNBR
004040 012737 014454 002242 MOV #EM3,ERRMSG
004046 012737 005426 002244 MOV #ERR4,ERRBLK
2329 004054 000261 SEC ;SET CARRY TO INDICATE ERROR
2330 004056 000401 BR 9$ ;EXIT WITH THE "ERROR" FLAG (CARRY BIT) SET
2331 004060 000241 5$: CLC ;CLEAR C BIT FOR NO ERRORS
2332 004062 000207 9$: RTS PC ;RETURN

```

M-LOOP - READ

```

2334 .SBTTL M-LOOP -- READ
2335 ;*****
2336 ; READ - READ THE SPECIFIED ADDRESS WITHIN THE DMV-11
2337 ;
2338 ; CALLING SEQUENCE:
2339 ;
2340 ; JSR R5,READ
2341 ; .WORD <ADDRESS OF REGISTER WITHIN DMV-11>
2342 ; .WORD <DESTINATION ADDRESS WITHIN LSI-11>
2343 ; ^CC N$ ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2344 ; ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
2345 ; <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
2346 ;
2347 ; N$: <RESUMPTION OF NORMAL PROCESSING>
2348 ;
2349 ;
2350 ;-----*****
2351 004064 012577 176272 READ: MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
2352 004070 112777 000001 176260 MOVB @REDLOC,@SEL2 ;TELL M LOOP TO GIVE US THE REQUESTED DATA
2353
2354 004076 010346 MOV R3,-(SP)
2355 004100 012703 000032 MOV @26.,R3 ;WAIT FOR THE M LOOP TO FINISH THE OPERATION
2356 004104 077301 1$: SOB R3,1$
2357 004106 012603 MOV (SP)+,R3
2358
2359 004110 132777 000200 176240 BITB @MRDY,@SEL2 ;DID THE M-LOOP FINISH
2360 004116 001023 BNE 5$ ;YES. GOOD. RETURN
2361
2362 004120 004737 004576 JSR PC,GETWSR ;GET BYTE SELECT REGISTERS
2363 004124 012737 000001 002310 MOV @REDLOC,GDATA ;IDENTIFY REQUESTED FUNCTION
2364 004132 GTDF EM4,ERR4 ;"MRDY" TIMEOUT
; QUEUE "DEVICE FATAL" ERROR # 3
;
; MOV #T.EDF,ERR1P
; MOV #3,ERRNBR
; MOV #EM4,ERRMSG
; MOV #ERR4,FRRBLK
004132 012737 000001 002236
004140 012737 000003 002240
004146 012737 014500 002242
004154 012737 005426 002244
2365 004162 000261 SEC ;INDICATE AN ERROR HAS BEEN STACKED
2366 004164 000401 BR 6$ ;RETURN WITH THAT INDICATION
2367
2368 004166 000241 5$: CLC ;INDICATE "NO ERROR"
2369 004170 117735 176172 6$: MOVB @SEL6,@(R5)+ ;PUT DATA WHERE CALLER WANTS IT
2370 004174 000205 RTS R5 ;RETURN

```

M-LOOP READ IMMEDIATE

```

2372 .SBTTL M-LOOP -- READ IMMEDIATE
2373 ;.....
2374 ; READI - READ IMMEDIATE THE SPECIFIED ADDRESS WITHIN THE DMV 11
2375 ;
2376 ; CALLING SEQUENCE:
2377 ;
2378 ;     JSR     R5,READI
2379 ;     .WORD  <ADDRESS OF REGISTER WITHIN DMV-11>
2380 ;     .WORD  <DESTINATION -- CONTENTS OF REG. IS PUT HERE>
2381 ;     BCC    N$          ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2382 ;     ERROR  N$          ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
2383 ;     <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
2384 ;
2385 ; N$:  <RESUMPTION OF NORMAL PROCESSING>
2386 ;
2387 ;-----
2388 ;
2389 004176 READI:
2390 004176 012577 176160      MOV     (R5),@SEL4      ;SETUP SOURCE POINTER
2391 004202 112777 000001 176146  MOVB   @REDLOC,@SEL2   ;TELL M-LOOP TO GIVE US THE REQUESTED DATA
2392
2393 004210 010346      MOV     R3,-(SP)
2394 004212 012703 000015      MOV     @13,R3         ;WAIT FOR THE M LOOP TO FINISH THE OPERATION
2395 004216 077301      SOB    R3,1$
2396 004220 012603      MOV     (SP),R3
2397
2398 004222 132777 000200 176126  BITB   @MRDY,@SEL2    ;DID THE M-LOOP FINISH
2399 004230 001023      BNE    5$             ;YES, GOOD. RETURN
2400
2401 004232 004737 004576      JSR    PC,GETWSR      ;GET BYTE SELECT REGISTERS
2402 004236 012737 000001 002310  MOV    @REDLOC,GDATA  ;IDENTIFY REQUESTED FUNCTION
2403 004244      GTDF   EM4,ERR4     ;"MRDY" TIMEOUT
2404      ;          QUEUE 'DEVICE FATAL' ERROR # 4
2405      MOV    @T,EDF,ERRTYP
2406      MOV    @4,ERRNBR
2407      MOV    @EM4,ERRMSG
2408      MOV    @ERR4,ERRBLK
2409      SEC
2410      BR    6$
2411
2412 5$:  CLC
2413 6$:  MOV    @SEL6,(R5)
2414      RTS   R5         ;INDICATE "NO ERROR"
2415      ;PUT DATA WHERE CALLER WANTS IT
2416      ;RETURN

```

M LOOP WRITE

2411
 2412
 2413
 2414
 2415
 2416
 2417
 2418
 2419
 2420
 2421
 2422
 2423
 2424
 2425
 2426
 2427
 2428 004310 012577 176046
 2429 004314 113577 176046
 2430 004320 000404

```

.SBTTL M LOOP -- WRITE
;*****
; WRITE - WRITE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
;
; CALLING SEQUENCE:
;
;     JSR     R5,WRITE
;     .WORD  <ADDRESS OF REGISTER WITHIN DMV-11>
;     .WORD  <ADDRESS OF DATA BYTE>
;     BCC   N$           ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
;     ERROR           ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
;     <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
;
; N$:  <RESUMPTION OF NORMAL PROCESSING>
;
;-----*****
WRITE:  MOV     (R5),@SEL4      ;SETUP SOURCE POINTER
        MOVB   @R5,@SEL6     ;MAKE DATA AVAILABLE TO M LOOP
        BR     MLWRI         ;THE REST OF THIS ROUTINE IS THE SAME AS "WRITEI

```

M LOOP WRITE IMMEDIATE

```

2432 .SBTTL M LOOP WRITE IMMEDIATE
2433 ;*****
2434 ; WRITEI - WRITE IMMEDIATE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
2435 ;
2436 ; CALLING SEQUENCE:
2437 ;
2438 ; JSR R5,WRITEI
2439 ; .WORD <ADDRESS OF REGISTER WITHIN DMV-11>
2440 ; .WORD <DATA FIELD -- DATA TO BE WRITTEN IN DMV-11>
2441 ; BCC N$ ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2442 ; ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
2443 ; <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
2444 ;
2445 ; N$: <RESUMPTION OF NORMAL PROCESSING>
2446 ;
2447 ;-----
2448
2449 004322 WRITEI:
2450 004322 012577 176034 MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
2451 004326 012577 176034 MOV (R5)+,@SEL6 ;MAKE DATA AVAILABLE TO M-LOOP
2452 004332 112777 0000C2 176016 MLWRI: MOVB @WRILOC,@SEL2 ;TELL M-LOOP TO WRITE THE DATA
2453
2454 004340 010346 MOV R3,-(SP)
2455 004342 012703 000050 MOV @40.,R3 ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2456 004346 077301 1$: SOB R3,1$
2457 004350 012603 MOV (SP)+,R3
2458
2459 004352 132777 000200 175776 BITB @MRDY,@SEL2 ;DID THE M-LOOP FINISH
2460 004360 001023 BNE 5$ ;YES, GOOD. RETURN
2461 004362 004737 004576 JSR PC,GETWSR ;GET BYTE SELECT REGISTERS
2462 004366 012737 000002 002310 MOV @WRILOC,GDATA ;IDENTIFY REQUESTED FUNCTION
2463 004374 GTDF EM4,ERR4 ;"MRDY" TIMEOUT
; QUEUE 'DEVICE FATAL' ERROR # 5
; MOV @T.EDF,ERRTYP
; MOV @S,ERRNBR
; MOV @EM4,ERRMSG
; MOV @ERR4,ERRBLK
004374 012737 000001 002236
004402 012737 000005 002240
004410 012737 014500 002242
004416 012737 005426 002244
2464 004424 000261 SEC ;INDICATE AN ERROR HAS BEEN STACKED
2465 004426 000401 BR 6$ ;RETURN WITH THAT INDICATION
2466
2467 004430 000241 5$: CLC ;INDICATE "NO ERROR"
2468 004432 000205 6$: RTS R5 ;RETURN

```


GETBSR GET BYTE SELECT REGISTERS

```

2470 .SBTTL GETBSR -- GET BYTE SELECT REGISTERS
2471 ;
2472 ;.....
2473 ;
2474 ; GET THE CONTENTS OF ALL CONTROL AND STATUS REGISTERS
2475 ;
2476 ; FUNCTION - THIS SUBROUTINE COLLECTS THE CONTENTS OF THE
2477 ; BYTE SELECT REGISTERS FOR THE PURPOSE OF DISPLAY.
2478 ;
2479 ; ENTRY CONDITIONS NONE 00 0 0000 0 00 0
2480 ; EXIT CONDITIONS - NONE 0 0 0 0 0 0 0 0
2481 ; REGISTERS DESTROYED NONE 00 0000 0000 0 0 0
2482 ;
2483 ;.....
2484 ;
2485 ;
2486 ;

```

```

2487 004434 117737 175712 002246 GETBSR: MOVB @BSSEL0,BSR0 ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
2488 004442 117737 175706 002250 MOVB @BSSEL1,BSR1 ;TABLE
2489 004450 117737 175702 002252 MOVB @BSSEL2,BSR2
2490 004456 117737 175676 002254 MOVB @BSSEL3,BSR3
2491 004464 117737 175672 002256 MOVB @BSSEL4,BSR4
2492 004472 117737 175666 002260 MOVB @BSSEL5,BSR5
2493 004500 117737 175662 002262 MOVB @BSSEL6,BSR6
2494 004506 117737 175656 002264 MOVB @BSSEL7,BSR7
2495 004514 117737 175652 002266 MOVB @BSSEL10,BSR10
2496 004522 117737 175646 002270 MOVB @BSSEL11,BSR11
2497 004530 117737 175642 002272 MOVB @BSSEL12,BSR12
2498 004536 117737 175636 002274 MOVB @BSSEL13,BSR13
2499 004544 117737 175632 002276 MOVB @BSSEL14,BSR14
2500 004552 117737 175626 002300 MOVB @BSSEL15,BSR15
2501 004560 117737 175622 002302 MOVB @BSSEL16,BSR16
2502 004566 117737 175616 002304 MOVB @BSSEL17,BSR17
2503 004574 000207 RTS PC ;RETURN TO CALLER
2504

```

```

2505 .SBTTL GETWSR - GET WORD SELECT REGISTERS
2506 ; "WORD" VERSION OF ABOVE SUBROUTINE
2507 ;
2508 004576 017737 175550 002246 GETWSR: MOV @WSEL0,WSR0 ;MOVE THE 8 WORD REGISTERS TO THE OTHERWISE
2509 004604 017737 175546 002250 MOV @WSEL2,WSR2 ;BYTE TABLE
2510 004612 017737 175544 002252 MOV @WSEL4,WSR4
2511 004620 017737 175542 002254 MOV @WSEL6,WSR6
2512 004626 017737 175540 002256 MOV @WSEL10,WSR10
2513 004634 017737 175536 002260 MOV @WSEL12,WSR12
2514 004642 017737 175534 002262 MOV @WSEL14,WSR14
2515 004650 017737 175532 002264 MOV @WSEL16,WSR16
2516 004656 000207 RTS PC ;RETURN TO CALLER

```

.INIT1 INITIALIZE TIMER # 1

2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574

004660 010146
004662 112537 002455
004666 112537 002457
004672 111537 002467
004676 142737 177477 002467
004704 012501

004706 106301
004710 042701 177677
004714 140177 175440
004720 106301
004722 052701 000100
004726 110137 002475

004732 004537 004310
004736 120016
004740 002475
004742 103431

004744 004537 004064
004750 120013
004752 002466
004754 103424

```

.SBTTL .INIT1 - INITIALIZE TIMER # 1
;*****
;* INITT1 - INITIALIZE TIMER # 1
;*
;* CALLING SEQUENCE:
;*
;* JSR R5,INITT1
;* .WORD <VALUE LOADED INTO THE T1 LATCH @ T1L & T1LH>
;* .WORD <BITS 6 & 7 WILL BE LOADED INTO "ACR", BIT 5 WILL BE
;* USED TO SET OR CLEAR BIT 6 ('T1') OF THE INTERRUPT
;* ENABLE REGISTER ("IER")>
;*
;* SEQUENCE OF EVENTS HEREIN:
;*
;* SET THE VIA'S INTERRUPT ENABLE REGISTER ("IER")
;*
;* SET THE VIA'S "ACR"
;*
;* SET T1L-L (ADDR 06)
;*
;* SET T1L-H (ADDR 07)
;*
;* RETURN WITHOUT ANY ERROR CHECKING
;*****
INITT1: MOV R1,-(SP) ;SAVE THE REGISTER WE WILL BE USING
        MOVB (R5),TMP6+1 ;SETUP VALUES TO BE LOADED INTO THE LATCHES
        MOVB (R5),TMP7+1
        MOVB (R5),TMPB+1 ;GET & PROCESS BITS FOR ACR 6 & 7
        BICB #C<BIT6+BIT7>,TMPB+1 ;EXTRACT BITS 6 & 7 & SAVE THEM FOR LATER
        MOV (R5),R1 ;NOW, GET THE BIT TO BE USED IN SETTING OR
        ;CLEARING BIT 6 OF 'IER"

; THE PASSED BIT IS IN THE WRONG POSITION BUT, IT SHOULD CONTROL THE OPERATION.
; WE KNOW WE ARE SETTING OR CLEARING BIT 6 - THUS, THE PASSED BIT WILL BECOME
; THE CONTROLLING BIT 7 AND WE WILL "OR" IN THE BIT WE WISH TO BE CONTROLLED
; (BIT 6).

        ASLB R1 ;THIS PUTS THE PASSED BIT INTO BIT 6.
        BIC #C<BIT6>,R1 ;WHILE HERE, CLEAR ALL OTHER BITS AND
        BICB R1,SEL3 ;CLEAR THE INTERRUPT FLAG IN THE SELECT REG.
        ASLB R1 ;NOW THE BIT IS IN THE CONTROLLING POSITION
        BIS #BIT6,R1 ;SET BIT 6
        MOVB R1,TMPE+1 ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

        JSR R5,WRITE ;WRITE TO
        IENR ;THE VIA'S IER
        TMPE+1 ;INTERRUPT ENABLE/DISABLE INFORMATION
        BCS 63$ ;EXIT ON ERROR

        JSR R5,RFAD ;READ THE CURRENT SETTING OF
        ACR ;THE VIA'S ACR
        TMPB
        BCS 63$ ;EXIT ON ERROR

```

.INIT1 INITIALIZE TIMER # 1

```

2575
2576 004756 013701 002466      MOV    TMPB,R1          ;GET THAT VALUE
2577 004762 042701 177477      BIC    #C<BIT6+BIT7>,R1 ;CLEAR BITS 6 & 7
2578 004766 150137 002467      BISB   R1,TMPB+1       ;ADD CURRENT BITS 0 -> 5 TO NEW BITS 6 & 7
2579
2580 004772 004537 004310      JSR    R5,WRITE        ;WRITE THE NEW REGISTER SETTING TO VIA-S ACR
2581 004776 120013
2582 005000 002467
2583 005002 103411      BCS    63$             ;EXIT ON ERROR
2584
2585 005004 004537 004310      JSR    R5,WRITE        ;WRITE TO
2586 005010 120006      T1LL
2587 005012 002455      TMP6+1                ;LOW ORDER LATCH REGISTER (T1L-L)
2588 005014 103404      BCS    63$             ;THE VALUE PASSED
2589
2590 005016 004537 004310      JSR    R5,WRITE        ;WRITE TO
2591 005022 120007      T1LH
2592 005024 002457      TMP7+1                ;HIGH ORDER LATCH REGISTER (T1L-H)
2593
2594
2595 005026 012601      63$: MOV    (SP)+,R1    ;RESTORE R1
2596 005030 000205      RTS    R5              ;RETURN
2597
2598
2599
2600      .SBTTL  STALL -- DELAY FOR 10.5 MICRO-SEC'S (ON LSI-11)
2601      ;*****
2602      ; STALL -- THIS SUBROUTINE STALLS FOR ABOUT 10.5 MICRO-SECONDS
2603      ;-----*****
2604 005032 000207      STALL: RTS    PC
2605

```

STREG STATIC TEST OF SPECIFIED DMV 11 LOCATION

```

2607 .SBTTL STREG -- STATIC TEST OF SPECIFIED DMV-11 LOCATION
2608
2609 ;*****
2610 ; STREG -- PERFORM A STATIC TEST OF THE SPECIFIED REGISTER
2611 ;
2612 ; CALLING SEQUENCE:
2613 ;
2614 ; <R0 CONTAINS THE ADDRESS OF THE REGISTER TO BE TESTED>
2615 ; <"TDATA" CONTAINS THE TEST BYTE>
2616 ; <"GDATA" CONTAINS THE EXPECTED DATA>
2617 ;
2618 ; JSR PC,STREG
2619 ; BCC N$ ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2620 ; ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
2621 ; <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
2622 ;
2623 ; N$: <RESUMPTION OF NORMAL PROCESSING>
2624 ;
2625 ; - *****
2626
2627 005034 010037 005050 STREG: MOV R0,2$ ;PUT SPECIFIED REGISTER'S ADDRESS IN I/O CALLS
2628 005040 010037 005062 MOV R0,4$
2629
2630 005044 004537 004310 2$: JSR R5,WRITE ;WRITE IT
2631 005050 000000 0 ;*** MODIFIED FROM ABOVE ***
2632 005052 002306 TDATA ;*** MODIFIED FROM ABOVE ***
2633 005054 103435 BCS 10$ ;ON ERROR, EXIT
2634
2635 005056 004537 004064 4$: JSR R5,READ ;READ IT BACK AGAIN
2636 005062 000000 0 ;*** MODIFIED FROM ABOVE ***
2637 005064 002312 BDATA
2638 005066 103430 BCS 10$ ;ON ERROR, EXIT
2639
2640 005070 123737 002310 002312 CMPB GDATA,BDATA ;DID WE READ WHAT WE WROTE?
2641 005076 000241 CLC ; (THIS ISN'T NEEDED FOR THE ERROR TEST BUT
2642 ; MUST BE CLEARED ON EXIT IF NO ERROR OCCURED)
2643 005100 001423 BEQ 10$ ;YES, EXIT FROM SUBTEST
2644 005102 013737 005050 002334 MOV 2$,REGNUM ;BUILD REGISTER #
2645 005110 042737 177760 002334 BIC #177760,REGNUM
2646 005116 GTDF EM25,ERR7 ;REPORT READ/WRITE ERROR
; QUEUE "DEVICE FATAL" ERROR # 6
; MOV #T.EDF,ERR7P
; MOV #6,ERRNBR
; MOV #EM25,ERRMSG
; MOV #ERR7,ERRBLK
005116 012737 000001 002236
005124 012737 000006 002240
005132 012737 015565 002242
005140 012737 006612 002244
2647 005146 000261 SEC
2648 005150 000207 10$: RTS PC ;INDICATE THAT AN ERROR WAS D:TECTED

```

INTERRUPT HANDLER MPIHAN

```

2650 .SBTTL INTERRUPT HANDLER -- MPIHAN
2651
2652 ;*****
2653 ; MPIHAN -- COUNT INTERRUPTS - USUALLY INTERRUPT "A"
2654 ;
2655 ; THIS ROUTINE WILL INCREMENT THE LOW BYTE OF "INTFLG" EACH TIME IT IS
2656 ; ENTERED. IF "IHILNK" IS NON ZERO, VECTOR TO THE ADDRESS THEREIN USING
2657 ; A "JSR PC"
2658 ; -*****
2659
2660 005152 BGNSRV MPIHAN
2661 005152 MPIHAN::
2662 005152 010046 MOV RO, (SP) ;SAVE RO
2663 005154 105737 002330 TSTB INTWCH ;HAVE WE BEEN TOLD TO WATCH FOR TYPE "A" INT'S?
2664 005160 001007 BNE 5$ ;YES, DO NORMAL INTERRUPT PROCESSING
2665 005162 004737 004434 JSR PC,GETBSR ;NO, DUMP REGISTERS AND
2666 005166 GDF EM34,ERR3 ; REPORT "UNEXPECTED INTERRUPT"
2667 ; "DEVICE FATAL" ERROR # 7
2668 ; TRAP C$ERDF
2669 ; .WORD 7
2670 ; .WORD EM34
2671 ; .WORD ERR3
2672 BR 10$ ;GO TO EXIT
2673
2674 5$: INCB INTFLG ;INCREMENT LOW BYTE OF INTERRUPT COUNTER
2675 TST IHILNK ;ARE WE EXPECTED TO EXECUTE ANOTHER ROUTINE?
2676 BEQ 10$ ;NO, GET OUT
2677 JSR PC,@IHILNK ;YES, GO TO IT I HOPE IT'S VALID!
2678 10$: MOV (SP)+,RO ;RESTORE RO
2679 ENDSRV ;RETURN TO INTERRUPTED PROCESS
2680 ; L10002:
2681 ; RTI
2682
2683 IHILNK: .WORD 0 ;POINTER TO AUXILIARY INT. HANDLING ROUTINE

```

INTERRUPT HANDLER MPOHAN

```

2677 .SBTTL INTERRUPT HANDLER -- MPOHAN
2678
2679 ;*****
2680 ; MPOHAN -- SIMPLY COUNT INTERRUPTS -- USUALLY INTERRUPT "B"
2681 ;
2682 ; THIS ROUTINE WILL INCREMENT THE HIGH BYTE OF "INTFLG" EACH TIME IT IS
2683 ; ENTERED. IF "IHOLNK" IS NON-ZERO, VECTOR TO THE ADDRESS THEREIN USING
2684 ; A "JSR PC"
2685 ;-----*****
2686
2687 005224 BGNSRV MPOHAN
2688 005224 MPOHAN::
2689 005224 010046 MOV RO, -(SP) ;SAVE RO
2690 005226 105737 002331 TSTB INTWCH+1 ;HAVE WE BEEN TOLD TO WATCH FOR TYPE "B" INT'S?
2691 005232 001007 BNE 5$ ;YES, DO NORMAL INTERRUPT PROCESSING
2692 005234 004737 004434 JSR PC,GETBSR ;NO, DUMP REGISTERS AND
2693 005240 GEDF EM34B,ERR3 ; REPORT "UNEXPECTED INTERRUPT"
; "DEVICE FATAL" ERROR # 8
; TRAP C$ERDF
; .WORD 8
; .WORD EM34B
; .WORD ERR3
2694 005240 104455 BR 10$ ;GO TO EXIT
2695 005242 000010
2696 005244 015644
2697 005246 005414
2698 005250 000407 5$: INCB INTFLG+1 ;INCREMENT HIGH BYTE OF INTERRUPT COUNTER
2699 005252 105237 002327 TST IHOLNK ;ARE WE EXPECTED TO EXECUTE ANOTHER ROUTINE?
2700 005256 005737 005274 BEQ 10$ ;NO, GET OUT
2701 005262 001402 JSR PC,@IHOLNK ;YES, GO TO IT - I HOPE IT S VALID!
2702 005264 004777 000004 10$: MOV (SP)+,RO ;RESTORE RO
2703 005270 012600 ENDSRV ;RETURN TO INTERRUPTED PROCESS
2704 005272 L10003:
2705 005274 000002 RTI
2706 2701 IHOLNK: .WORD 0 ;POINTER TO AUXILIARY INT. HANDLING ROUTINE
2707 2702

```

GLOBAL ERROR REPORT REPORT SECTION

```

2704 .SBTTL GLOBAL ERROR REPORT REPORT SECTION
2705
2706 ;////////////////////////////////////
2707 ;/ THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
2708 ;/ THAT ARE USED IN MORE THAN ONE TEST.
2709 ;////////////////////////////////////
2710 .EVEN
2711
2712 ;-----
2713 ;SBTTL ERROR HANDLER -- ERR1 - "NO NOTHING" HANDLER
2714 ;-----
2714 005276 BGNMSG ERR1
2715 005276 004737 012072 JSR PC,NULERR ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
2716 005302 ENDMSG
2717 005302 104423 L10004: TRAP C$MSG
2718
2719 ;-----
2720 ;SBTTL ERROR HANDLER -- ERR2 - CSR REGISTER ERROR REPORTING
2721 ;-----
2720 005304 BGNMSG ERR2
2721 005304 PRINTB @FMT02,@TXT5,REGNUM ERR2::
2721 005304 013746 002334 MOV REGNUM,-(SP)
2721 005310 012746 013605 MOV @TXT5,-(SP)
2721 005314 012746 012124 MOV @FMT02,(SP)
2721 005320 012746 000003 MOV @3,-(SP)
2721 005324 010600 MOV SP,RO
2721 005326 104414 TRAP C$PNTB
2722 005330 062706 000010 ADD @10,SP
2723 005334 004737 011276 JSR PC,XORGB
2723 005340 PRINTB @FMT02A,<B,GDATA>,<B,EData>,<B,XDATA>
2723 005340 005046 CLR -(SP)
2723 005342 153716 002314 BISB XDATA,(SP)
2723 005346 005046 CLR -(SP)
2723 005350 153716 002312 BISB BDATA,(SP)
2723 005354 005046 CLR -(SP)
2723 005356 153716 002310 BISB GDATA,(SP)
2723 005362 012746 012161 MOV @FMT02A,-(SP)
2723 005366 012746 000004 MOV @4,-(SP)
2723 005372 010600 MOV SP,RO
2723 005374 104414 TRAP C$PNTB
2723 005376 062706 000012 ADD @12,SP
2724 005402 004737 011322 JSR PC,ERR4$ ;DUMP THE BYTE SELECT REGISTERS
2725 005406 004737 012072 JSR PC,NULERR ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
2726 005412 ENDMSG
2726 005412 104423 L10005: TRAP C$MSG
2727
2728 ;-----
2729 ;SBTTL ERROR HANDLER ERR3 DUMP THE BYTE SELECT REGISTERS
2730 ;-----
2730 005414 BGNMSG ERR3
2731 005414 ERR3::
2731 005414 004737 011322 JSR PC,ERR4$
2732 005420 004737 012072 JSR PC,NULERR ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
2733 005424 ENDMSG
2733 005424 104423 L10006: TRAP C$MSG

```

ERROR HANDLER ERR3 DUMP THE BYTE SELECT REGISTERS

```

2734
2735      ;SBTTL  ERROR HANDLER  ERR4  M-LOOP TIMEOUT ERROR HANDLING
2736      ;-----
2737      BGNMSG  ERR4
2738      005426      010146      MOV      R1,-(SP)      ;SAVE THE WORKING REGISTER
2739      005430      113701      002310      MOV      GDATA,R1      ;SAVE THIS FOR LATER
2740      005434      122701      000017      CMP      #17,R1      ;WAS THIS AN M-LOOP REQUEST?
2741      005440      103013      BHS      5$      ;YES, THEN REPORT THE FUNCTION CODE
2742      005442      005046      PRINTX  #FMT5,<B,R1>      ;NO, THEN IT MUST BE A BSEL1 SETTING
2743      005466      000425      BR      20$
2744
2745      005470      001001      5$:      BNE      6$      ;IF IT WAS A 17, THIS IS A "NOP" AND
2746      005472      005001      CLR      R1      ; THE TEXT POINTER MUST SO REFLECT.
2747      005474      022701      000007      6$:      CMP      #7,R1      ;IS FUNCTION CODE > 7?
2748      005500      003002      BGT      7$      ;NO, THEN WE CAN HANDLE IT
2749      005502      012701      000006      MOV      #6,R1      ;YES, THEN IT'S UNDEFINED SAY SO
2750      005506      006301      7$:      ASL      R1      ;CONVERT TO A WORD OFFSET
2751      005510      016146      017532      PRINTX  #FMT5A,<B,GDATA>,TXTMLT(R1) ;REPORT THE FAILING FUNCTION
2752      005510      016146      017532      MOV      TXTMLT(R1),(SP)
2753      005514      005046      CLR      -(SP)
2754      005516      153716      002310      BISB    GDATA,(SP)
2755      005522      012746      012443      MOV      #FMT5A,-(SP)
2756      005526      012746      000003      MOV      #3,(SP)
2757      005532      010600      MOV      SP,RO
2758      005534      104415      TRAP    C$PNTX
2759      005536      062706      000010      ADD     #10,SP
2760
2761      005542      012601      20$:      MOV      (SP)+,R1      ;RESTORE THE WORKING REGISTER
2762      005544      004737      011710      JSR      PC,ERR5$      ;DUMP THE SELECT REGISTERS
2763      005550      ENDMSG
2764      005550      104423      L10007:  TRAP    C$MSG

```

```

2756      ;-----
2757      ;SBTTL  ERROR HANDLER  ERR5  - WORD SELECT REG. ERRORS
2758      ;-----
2759      BGNMSG  ERR5
2760      005552      013746      002334      PRINTB  #FMT02,#TXT5,REGNUM
2761      005552      013746      002334      MOV      REGNUM,-(SP)
2762      005556      012746      013605      MOV      #TXT5,-(SP)
2763      005562      012746      012124      MOV      #FMT02,-(SP)
2764      005566      012746      000003      MOV      #3,-(SP)
2765      005572      010600      MOV      SP,RO
2766      005574      104414      TRAP    C$PNTB
2767      005576      062706      000010      ADD     #10,SP
2768
2769      005602      004737      011276      JSR      PC,XORGB
2770      005606      013746      002314      PRINTB  #FMT10,GDATA,BDATA,XDATA
2771      005612      013746      002312      MOV      XDATA,-(SP)
2772      005612      013746      002312      MOV      BDATA,-(SP)

```


ERROR HANDLER -- ERR5 -- WORD SELECT REG. ERRORS

```

005616 013746 002310          MOV      GDATA,-(SP)
005622 012746 012654          MOV      #FMT10,-(SP)
005626 012746 000004          MOV      #4,-(SP)
005632 010600          MOV      SP,RO
005634 104414          TRAP     C$PNTB
005636 062706 000012          ADD      #12,SP
2763 005642 004737 011710          JSR      PC,ERR5$          ;DUMP THE SELECT REGISTERS
2764 005646          ENDMMSG
                                L10010:
005646          TRAP     C$MSG
005646 104423
2765          ;-----
2766          ;SBTTL ERROR HANDLER -- ERR6 -- VIA REGISTER ERRORS W/FULL REG. DUMP
2767          ;-----
2768 005650          BGNMSG  ERR6
005650          ERR6::
2769          ;*** PRINT THE FIRST HALF OF THE REGISTERS ***
2770 005650 010146          MOV      R1,-(SP)          ;PRESERVE R1'S CONTENTS
2771 005652 012701 002604          MOV      #PATCR,R1          ;POINT TO EXPECTED VALUES
2772 005656          PRINTX #FMT06,#TXT7
                                MOV      #TXT7,-(SP)
                                MOV      #FMT06,-(SP)
                                MOV      #2,-(SP)
                                MOV      SP,RO
                                TRAP     C$PNTX
                                ADD      #6,SP
2773 005702          PRINTX #FMT06A,#TXT8A,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>
005702 005046          CLR      -(SP)
005704 152116          BISB    (R1)+,(SP)
005706 005046          CLR      -(SP)
005710 152116          BISB    (R1)+,(SP)
005712 005046          CLR      -(SP)
005714 152116          BISB    (R1)+,(SP)
005716 005046          CLR      -(SP)
005720 152116          BISB    (R1)+,(SP)
005722 005046          CLR      -(SP)
005724 152116          BISB    (R1)+,(SP)
005726 005046          CLR      -(SP)
005730 152116          BISB    (R1)+,(SP)
005732 012746 014004          MOV      #TXT8A,-(SP)
005736 012746 012571          MOV      #FMT06A,-(SP)
005742 012746 000010          MOV      #10,-(SP)
005746 010600          MOV      SP,RO
005750 104415          TRAP     C$PNTX
005752 062706 000022          ADD      #22,SP
2774 005756          PRINTX #FMT06B,<B,(R1)>,<B,(R1)>
                                CLR      -(SP)
                                BISB    (R1)+,(SP)
                                CLR      -(SP)
                                BISB    (R1)+,(SP)
                                MOV      #FMT06B,(SP)
                                MOV      #3,(SP)
                                MOV      SP,RO
                                TRAP     C$PNTX
                                ADD      #10,SP
005756 005046          CLR      -(SP)
005760 152116          BISB    (R1)+,(SP)
005762 005046          CLR      -(SP)
005764 152116          BISB    (R1)+,(SP)
005766 012746 012637          MOV      #FMT06B,(SP)
005772 012746 000003          MOV      #3,(SP)
005776 010600          MOV      SP,RO
006000 104415          TRAP     C$PNTX
006002 062706 000010          ADD      #10,SP
2775 006006 012701 003122          MOV      #BT1,R1          ;POINT TO ACTUAL VALUES
2776 006012          PRINTX #FMT06A,#TXT8E,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>
006012 005046          CLR      (SP)

```

ERROR HANDLER -- ERR6 -- VIA REGISTER ERRORS W/FULL REG. DUMP

SEQ 0065

006014	152116				BISB	(R1)+,(SP)
006016	005046				CLR	-(SP)
006020	152116				BISB	(R1)+,(SP)
006022	005046				CLR	-(SP)
006024	152116				BISB	(R1)+,(SP)
006026	005046				CLR	-(SP)
006030	152116				BISB	(R1)+,(SP)
006032	005046				CLR	-(SP)
006034	152116				BISB	(R1)+,(SP)
006036	005046				CLR	-(SP)
006040	152116				BISB	(R1)+,(SP)
006042	012746	014021			MOV	#TXT8B,-(SP)
006046	012746	012571			MOV	#FMT06A,-(SP)
006052	012746	000010			MOV	#10,-(SP)
006056	010600				MOV	SP,RO
006060	104415				TRAP	C\$PNTX
2777 006062	062706	000022			ADD	#22,SP
006066			PRINTX	#FMT06B,<B,(R1)+>,<B,(R1)+>		
006066	005046				CLR	-(SP)
006070	152116				BISB	(R1)+,(SP)
006072	005046				CLR	-(SP)
006074	152116				BISB	(R1)+,(SP)
006076	012746	012637			MOV	#FMT06B,-(SP)
006102	012746	000003			MOV	#3,-(SP)
006106	010600				MOV	SP,RO
006110	104415				TRAP	C\$PNTX
2778 006112	062706	000010			ADD	#10,SP
2779 006116	012701	003206	MOV	#B12,R1 ;POINT TO XOR VALUES		
006122			PRINTX	#FMT06A,#TXT8C,<B,(R1)+>,<B,(R1)+>,<B,(R1)+>,<B,(R1)+>,<B,(R1)+>,<B,(R1)+>,<B,(R1)+>		
006122	005046				CLR	-(SP)
006124	152116				BISB	(R1)+,(SP)
006126	005046				CLR	-(SP)
006130	152116				BISB	(R1)+,(SP)
006132	005046				CLR	-(SP)
006134	152116				BISB	(R1)+,(SP)
006136	005046				CLR	-(SP)
006140	152116				BISB	(R1)+,(SP)
006142	005046				CLR	-(SP)
006144	152116				BISB	(R1)+,(SP)
006146	005046				CLR	-(SP)
006150	152116				BISB	(R1)+,(SP)
006152	012746	014036			MOV	#TXT8C,-(SP)
006156	012746	012571			MOV	#FMT06A,-(SP)
006162	012746	000010			MOV	#10,(SP)
006166	010600				MOV	SP,RO
006170	104415				TRAP	C\$PNTX
2780 006172	062706	000022			ADD	#22,SP
006176			PRINTX	#FMT06B,<B,(R1)+>,<B,(R1)+>		
006176	005046				CLR	-(SP)
006200	152116				BISB	(R1)+,(SP)
006202	005046				CLR	-(SP)
006204	152116				BISB	(R1)+,(SP)
006206	012746	012637			MOV	#FMT06B,(SP)
006212	012746	000003			MOV	#3,(SP)
006216	010600				MOV	SP,RO
006220	104415				TRAP	C\$PNTX
006222	062706	000010			ADD	#10,SP

ERROR HANDLER - ERR6 VIA REGISTER ERRORS W/FULL REG. DUMP

```

2781
2782 006226 012701 002614      ;*** PRINT SECOND HALF OF THE REGISTERS ***
2783 006232                                MOV     #PACR.8.,R1      ;POINT TO 2ND HALF OF REGISTERS EXPECTED VALUES
                                PRINTX  #FMT06,#TXT7A
                                MOV     #TXT7A,(SP)
                                MOV     #FMT06,(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C:PNTX
                                ADD     #6,SP
2784 006252 062706 000006      PRINTX  #FMT06A,#TXT8A,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     (SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     (SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                MOV     #TXT8A,-(SP)
                                MOV     #FMT06A,-(SP)
                                MOV     #10,-(SP)
                                MOV     SP,R0
                                TRAP    C:PNTX
                                ADD     #22,SP
2785 006326 062706 000022      PRINTX  #FMT06B,<B,(R1)>,<B,(R1)>
                                CLR     (SP)
                                BISB   (R1),-(SP)
                                CLR     (SP)
                                BISB   (R1),-(SP)
                                MOV     #FMT06B,-(SP)
                                MOV     #3,-(SP)
                                MOV     SP,R0
                                TRAP    C:PNTX
                                ADD     #10,SP
2786 006352 062706 000010
2787 006362 012701 003132      MOV     #BT1.8.,R1      ;POINT TO 2ND HALF OF ACTUAL VALUES
                                PRINTX  #FMT06A,#TXT8B,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                CLR     -(SP)
                                BISB   (R1),-(SP)
                                MOV     #TXT8B,(SP)
                                MOV     #FMT06A,(SP)
                                MOV     #10,-(SP)
                                MOV     SP,R0
                                TRAP    C:PNTX
006232 012746 013717
006236 012746 012562
006242 012746 000002
006246 010600
006250 104415
006256 005046
006260 152116
006262 005046
006264 152116
006266 005046
006270 152116
006272 005046
006274 152116
006276 005046
006300 152116
006302 005046
006304 152116
006306 012746 014004
006312 012746 012571
006316 012746 000010
006322 010600
006324 104415
006332 005046
006334 152116
006336 005046
006340 152116
006342 012746 012637
006346 012746 000003
006352 010600
006354 104415
006356 062706 000010
006400 152116
006402 005046
006404 152116
006406 005046
006410 152116
006412 005046
006414 152116
006416 012746 014021
006422 012746 012571
006426 012746 000010
006432 010600
006434 104415

```

ERROR HANDLER - ERR6 VIA REGISTER ERRORS W/FULL REG. DUMP

SEQ 0067

```

2788 006436 062706 000022
006442 PRINTX @FMT06B,<B,(R1)>,<B,(R1)>
006442 005046
006444 152116
006446 005046
006450 152116
006452 012746 012637
006456 012746 000003
006462 010600
006464 104415
006466 062706 000010
2789 006472 012701 003216
2790 006476 MOV @BT2+8.,R1 ;POINT TO 2ND HALF OF XOR VALUES
006476 PRINTX @FMT06A,@TXT8C,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>,<B,(R1)>
006476 005046
006500 152116
006502 005046
006504 152116
006506 005046
006510 152116
006512 005046
006514 152116
006516 005046
006520 152116
006522 005046
006524 152116
006526 012746 014036
006532 012746 012571
006536 012746 000010
006542 010600
006544 104415
006546 062706 000022
2791 006552 PRINTX @FMT06B,<B,(R1)>,<B,(R1)>
006552 005046
006554 152116
006556 005046
006560 152116
006562 012746 012637
006566 012746 000003
006572 010600
006574 104415
006576 062706 000010
2792 006602 012601
2793 006604 004737 012072
2794 006610
006610
006610 104423
2795
2796
2797
2798 006612
006612
2799 006612 113701 002334
2800 006616 006301
2801 006620
006620 016146 017554
006624 012746 014327
006630 012746 012530

```

```

ADD @22,SP
CLR -(SP)
BISB (R1),-(SP)
CLR -(SP)
BISB (R1),-(SP)
MOV @FMT06B, (SP)
MOV @3, (SP)
MOV SP,R0
TRAP C$PNTX
ADD @10,SP
CLR -(SP)
BISB (R1),-(SP)
CLR -(SP)
BISB (R1),-(SP)
CLR (SP)
BISB (R1),-(SP)
CLR -(SP)
BISB (R1),-(SP)
CLR (SP)
BISB (R1),-(SP)
CLR -(SP)
BISB (R1),-(SP)
MOV @TXT8C, -(SP)
MOV @FMT06A, -(SP)
MOV @10, -(SP)
MOV SP,R0
TRAP C$PNTX
ADD @22,SP
CLR (SP)
BISB (R1),-(SP)
CLR -(SP)
BISB (R1),-(SP)
MOV @FMT06B, -(SP)
MOV @3, -(SP)
MOV SP,R0
TRAP C$PNTX
ADD @10,SP
MOV (SP),R1 ;RESTORE R1
JSR PC,NULERR ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
ENDMSG
L10011:
TRAP C$MSG

```

```

:SBTTL ERROR HANDLER -- ERR7 VIA REGISTER ERRORS
:-----
BGNMSG ERR7
ERR7::
MOV# REGNUM,R1
ASL R1 ;AS PASSED, THIS WAS A BYTE OFFSET
PRINTB @FMT07,@TXIVR, TXIVRT(R1)
MOV TXIVRT(R1), -(SP)
MOV @TXIVR, -(SP)
MOV @FMT07, (SP)

```

ERROR HANDLER ERR7 - VIA REGISTER ERRORS

```

006634 012746 000003
006640 010600
006642 104414
006644 062706 000010
2802 006650 004737 011276 JSR PC,XORGB
2803 006654 PRINTB #FMT02A,<B,GDATA>,<B,BDATA>,<B,XDATA>
006654 005046
006656 153716 002314
006662 005046
006664 153716 002312
006670 005046
006672 153716 002310
006676 012746 012161
006702 012746 000004
006706 010600
006710 104414
006712 062706 000012
2804 006716 004737 012072 JSR PC,NULERR ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
2805 006722 ENDMMSG
006722 104423 L10012:
006722 TRAP C$MSG
2806
2807 ;-----
2808 ;SBTTL ERROR HANDLER -- ERR47 FOR RAM DATA ERRORS IN STATIC TEST(S)
2809 ;-----
006724 BGNMSG ERR47
006724 ERR47::
2810 ; PRINT HEADING LINE # 1
2811
2812 006724 013700 002444 MOV TMP2,RO ;GET TEST PATTERN CODE
2813 006730 001404 BEQ 2# ;ZERO IS UNDEFINED BUT THERE IS TEXT TO SAY THAT
2814 006732 020027 000006 CMP RO,#6 ;THIS IT ALL WE UNDERSTAND FOR NOW
2815 006736 003401 BLE 2# ;IF WITHIN LIMITS, LET IT GO
2816 006740 005000 CLR RO ;ELSE, MAKE IT 0 FOR "UNDEFINED"
2817 006742 006300 2# ASL RO ;CONVERT TO A WORD INDEX
2818 006744 016000 007614 MOV TXT47P(RO),RO ;GET ADDRESS OF REQUIRED TEXT
2819 006750 PRINTX #FMT47A,RO ;IDENTIFY TEST PATTERN BEING USED
006750 010046 MOV RO,(SP)
006752 012746 007232 MOV #FMT47A,-(SP)
006756 012746 000002 MOV #2,(SP)
006762 010600 MOV SP,RO
006764 104415 TRAP C$PNTX
006766 062706 000006 ADD #6,SP
2820 ; PRINT HEADING LINE # 2
2821
2822 006772 PRINTX #FMT47B ;STANDARD PORTION OF LINE 2
006772 012746 007263 MOV #FMT47B,-(SP)
006776 012746 000001 MOV #1,(SP)
007002 010600 MOV SP,RO
007004 104415 TRAP C$PNTX
007006 062706 000004 ADD #4,SP
2823 ; PRINT HEADING LINE # 3
2824
2825 007012 PRINTX #FMT47C ;STANDARD PORTION OF LINE 3
007012 012746 007322 MOV #FMT47C,-(SP)
007016 012746 000001 MOV #1,(SP)
007022 010600 MOV SP,RO
007024 104415 TRAP C$PNTX

```

ERROR HANDLER - ERR47 - FOR RAM DATA ERRORS IN STATIC TEST(S)

```

2826 007026 062706 000004          ;          PRINT HEADING LINE # 4          ADD      #4,SP
2827                                     ;
2828 007032                                     PRINTX  #FMT47E          ;STANDARD PORTION OF LINE 4
007032 012746 007350          MOV      #FMT47E, -(SP)
007036 012746 000001          MOV      #1, -(SP)
007042 010600          MOV      SP,RO
007044 104415          TRAP    C$PNTX
007046 062706 000004          ADD      #4,SP
2829                                     ;          GO PRINT DATA PORTION OF ERROR MESSAGE
2830                                     ;
2831 007052                                     PRINTX  #NEWLIN          ;TERMINATE HEADER & CAUSE 1 BLANK LINE
007052 012746 012121          MOV      #NEWLIN, (SP)
007056 012746 000001          MOV      #1, -(SP)
007062 010600          MOV      SP,RO
007064 104415          TRAP    C$PNTX
007066 062706 000004          ADD      #4,SP
2832 007072 005037 007104          CLR     ER47CT          ;RE-INITIALIZE THE DATA LINE COUNTER
2833 007076 004737 007110          JSR     PC,ERR47.      ;USE COMMON SUBROUTINE TO REPORT DATA
2834 007102                                     ENOMSG
007102                                     L10013:
007102 104423                                     TRAP    C$MSG
2835                                     ;
2836 007104 000000          ER47CT: .WORD 0          ;THIS VARIABLE WILL COUNT THE DATA LINES
2837 007106 000020          ER47MX: .WORD 16.      ;THIS CONSTANT LIMITS THE DATA LINES PRINTED
2838                                     ;
2839 007110          ERR47.:
2840                                     ;
2841 007110 023737 007104 007106          CMP     ER47CT,ER47MX  ;HAVE WE REPORTED ENOUGH OF THESE DATA LINES?
2842 007116 103044          BHS    60$            ;YES, BYPASS THIS WHOLE ROUTINE AND EXIT
2843 007120 005237 007104          INC     ER47CT        ;NO, COUNT THIS LINE
2844                                     ;
2845 007124 113701 002450          MOVB   TMP4,R1        ;GET EXPECTED DATA
2846 007130 113703 002452          MOVB   TMP5,R3        ;SETUP TO CALCULATE XOR
2847 007134 074103          XOR    R1,R3          ;CALCULATE XOR OF EXPECTED & ACTUAL DATA
2848 007136          PRINTX #FMT47G,TMP4,<B,R1>,<B,TMP5>,<B,R3> ;PRINT DATA LINE
007136 005046          CLR     -(SP)
007140 150316          BISB   R3,(SP)
007142 005046          CLR     -(SP)
007144 153716 002452          BISB   TMP5,(SP)
007150 005046          CLR     -(SP)
007152 150116          BISB   R1,(SP)
007154 013746 002464          MOV    TMP4, -(SP)
007160 012746 007407          MOV    #FMT47G, -(SP)
007164 012746 000005          MOV    #5, (SP)
007170 010600          MOV    SP,RO
007172 104415          TRAP   C$PNTX
007174 062706 000014          ADD    #14,SP
2849 007200 023737 007104 007106          CMP     ER47CT,ER47MX  ;IF THESE TWO ARE EQUAL, WE WON'T BE PRINTING
2850 007206 001010          BNE    60$            ;ANY MORE LINES FOR A WHILE. SO,
2851 007210          PRINTX #FMT48I          ; PUT OUT A MESSAGE TO THAT EFFECT.
007210 012746 010644          MOV    #FMT48I, (SP)
007214 012746 000001          MOV    #1, -(SP)
007220 010600          MOV    SP,RO
007222 104415          TRAP   C$PNTX
007224 062706 000004          ADD    #4,SP
2852 007230 000207          60$:   RTS     PC

```

ERROR HANDLER - ERR47 -- FOR RAM DATA ERRORS IN STATIC TEST(S

```

2853
2854 .NLIST BEX
2855 007232 045 116 045 FMT47A: .ASCIZ \N#S2#ATEST PATTERN: #T\
2856 007263 045 116 045 FMT47B: .ASCIZ \N#S2#A (ALL VALUES IN OCTAL)\
2857 007322 045 116 045 FMT47C: .ASCIZ \N#S3#A RAM SHOULD\
2858 007350 045 116 045 FMT47E: .ASCIZ \N#S3#ADDRESS BE IS XOR\
2859 007407 045 116 045 FMT47G: .ASCIZ \N#S4#04#S4#03#S3#03#S2#03\
2860 007442 101 114 114 TXT47C: .ASCIZ \ALL ONES\
2861 007453 101 114 114 TXT47D: .ASCIZ \ALL ZEROES\
2862 007466 061 040 102 TXT47E: .ASCIZ \1 BIT ALTERNATING\
2863 007510 062 040 102 TXT47F: .ASCIZ \2 BITS ALTERNATING\
2864 007533 101 104 104 TXT47G: .ASCIZ \ADDRESS IN ADDRESS\
2865 007556 111 116 103 TXT47H: .ASCIZ \INCREMENTAL VALUE IN ADDRESS\
2866 .LIST BEX
2867 .EVEN
2868 007614 014235 007442 007453 TXT47P: .WORD TXTML6,TXT47C,TXT47D,TXT47E,TXT47F,TXT47G,TXT47H
007622 007466 007510 007533
007630 007556

```

```

2869
2870 ; "TXTML6" ABOVE IS DEFINED AS "UNDEFINED" IN THE M-LOOP FUNCTION DEF'S.
2871
2872
2873 :-----
2874 .SBTTL ERROR HANDLER - ERR48 - FOR DATA ERRORS IN 'MOVING INVERSIONS TEST'
2875 :-----

```

```

2875 007632 BGNMSG ERR48
007632 ERR48::
2876 ; PRINT HEADING LINE # 1
2877
2878 007632 PRINTX #FMT48A ;STANDARD PORTION OF LINE 1
007632 012746 010300 MOV #FMT48A, -(SP)
007636 012746 000001 MOV #1, -(SP)
007642 010600 MOV SP,RO
007644 104415 TRAP C$PNTX
007646 062706 000004 ADD #4,SP
2879 007652 032737 000004 002350 BIT #BIT2,PFLAG ;IF EXTENDED INFORMATION REQUESTED.
2880 007660 001410 BEQ 2$
2881 007662 PRINTX #FMT48B ;PRINT EXTENDED PORTION OF LINE 1
007662 012746 010347 MOV #FMT48B, (SP)
007666 012746 000001 MOV #1, (SP)
007672 010600 MOV SP,RO
007674 104415 TRAP C$PNTX
007676 062706 000004 ADD #4,SP
2882 ; PRINT HEADING LINE # 2
2883
2884 2$: PRINTX #FMT48C ;STANDARD PORTION OF LINE 2
007702 012746 010402 MOV #FMT48C, (SP)
007706 012746 000001 MOV #1, (SP)
007712 010600 MOV SP,RO
007714 104415 TRAP C$PNTX
007716 062706 000004 ADD #4,SP
2885 ; PRINT HEADING LINE # 3
2886
2887
2888 PRINTX #FMT48E ;STANDARD PORTION OF LINE 3
007722 012746 010437 MOV #FMT48E, (SP)
007726 012746 000001 MOV #1, (SP)
007732 010600 MOV SP,RO

```

ERROR HANDLER - ERR48 - FOR DATA ERRORS IN "MOVING INVERSION"

SEQ 0071

```

007734 104415
007736 062706 000004
2889 007742 032737 000004 002350 BIT #BIT2,PFLAG ;IF EXTENDED INFORMATION REQUESTED.
2890 007750 001410 BEQ 6$
2891 007752 PRINTX #FMT48F ;PRINT EXTENDED PORTION OF LINE 3
007752 012746 010506 MOV #FMT48F,-(SP)
007756 012746 000001 MOV #1,(SP)
007762 010600 MOV SP,R0
007764 104415 TRAP C$PNTX
007756 062706 000004 ADD #4,SP
2892 ; GO PRINT DATA PORTION OF ERROR MESSAGE
2893
2894 007772 6$: PRINTX #NEWLIN ;TERMINATE HEADER & CAUSE 1 BLANK LINE
007772 012746 012121 MOV #NEWLIN,-(SP)
007776 012746 000001 MOV #1,-(SP)
010002 010600 MOV SP,R0
010004 104415 TRAP C$PNTX
010006 062706 000004 ADD #4,SP
2895 010012 005037 010024 CLR ER48CT ;RE-INITIALIZE THE DATA LINE COUNTER
2896 010016 004737 010030 JSR PC,ERR48. ;USE COMMON SUBROUTINE TO REPORT DATA
2897 010022 ENDMSG
010022 104423 L10014:
010022 TRAP C$MSG
2898
2899 010024 000000 ER48CT: .WORD 0 ;THIS VARIABLE WILL COUNT THE DATA LINES
2900 010026 000020 ER48MX: .WORD 16. ;THIS CONSTANT LIMITS THE DATA LINES PRINTED
2901
2902 010030 ERR48.:
2903
2904 010030 023737 010024 010026 CMP ER48CT,ER48MX ;HAVE WE REPORTED ENOUGH OF THESE DATA LINES?
2905 010036 103117 BHIS 60$ ;YES, BYPASS THIS WHOLE ROUTINE AND EXIT
2906 010040 005237 010024 INC ER48CT ;NO, COUNT THIS LINE
2907
2908 ; DETERMINT WHICH ERROR CALL GOT US HERE PRE-WRITE OR POST-WRITE:
2909
2910 010044 032737 000002 002476 BIT #BIT1,TMPF ;DID PRE-WRITE ERROR CALL GET US HERE?
2911 010052 001405 BEQ 2$ ;NO, THEN SETUP FOR "POST" IN ERROR MESSAGE
2912 010054 012700 010736 MOV #TXT48A,R0 ;YES, SETUP FOR "PRE" IN ERROR MESSAGE
2913 010060 113701 002450 MOVB TMP4,R1 ;GET EXPECTED DATA (BEFORE WRITING NEW VALUE)
2914 010064 000404 BR 4$
2915
2916 010066 012700 010743 2$: MOV #TXT48B,R0 ;POINT TO "POST" TEXT
2917 010072 113701 002451 MOVB TMP4+1,R1 ;GET EXPECTED DATA (AFTER WRITING NEW VALUE)
2918 010076 013703 002452 4$: MOV TMP5,R3 ;SETUP TO CALCULATE XOR
2919 010102 074103 XOR R1,R3 ;CALCULATE XOR OF EXPECTED & ACTUAL DATA
2920 010104 PRINTX #FMT48G,R0,TMPA,<B,R1>,<B,TMP5> <3> ;PRINT STANDARD DATA LINE
010104 005046 CLR (SP)
010106 150316 BISB R3,(SP)
010110 005046 CLR -(SP)
010112 153716 002452 BISB TMP5,(SP)
010116 005046 CLR (SP)
010120 150116 BISB R1,(SP)
010122 013746 002464 MOV TMPA,-(SP)
010126 010046 MOV R0,(SP)
010130 012746 010551 MOV #FMT48G,-(SP)
010134 012746 000006 MOV #6,(SP)
010140 010600 MOV SP,R0

```


ERROR HANDLER - ERR48 - FOR DATA ERRORS IN "MOVING INVERSION"

```

010142 104415
010144 062706 000016
2921 010150 032737 000004 002350 BIT #BIT2,PFLAG ;IF EXTENDED INFORMATION REQUESTED,
2922 010156 001433 BEQ 10$
2923 ;SETUP FOR PRINTING OF EXTENDED INFORMATION
2924 010160 013701 002470 MOV TMPD,R1 ;DATA BIT VALUE (0 OR 1)
2925 010164 042701 177776 BIC #+CBIT0,R1 ; MAKE SURE WE ONLY HAVE ONE BIT
2926 010170 005737 002472 TST TMPD ;DIRECTION?
2927 010174 001003 BNE 6$ ;BACKWARD -
2928 010176 012700 010750 MOV #TXT48C,R0 ;FORWARD
2929 010202 000402 BR 8$
2930 010204 012700 010755 6$: MOV #TXT48D,R0 ;BACKWARD
2931 010210 8$: PRINTX #FMT48H,TMPB,R1,R0,<B,TMP9> ;PRINT EXTENDED INFORMATION
010210 005046 CLR -(SP)
010212 153716 002462 BISB IM),(CF)
010216 010046 MOV R0,-(SP)
010220 010146 MOV R1,(SP)
010222 013746 002466 MOV TMPB,(SP)
010226 012746 010611 MOV #FMT48H,(SP)
010232 012746 000005 MOV #5,-(SP)
010236 010600 MOV SP,R0
010240 104415 TRAP C$PNTX
010242 062706 000014 ADD #14,SP
2932 010246 023737 010024 010026 10$: CMP ER48CT,ER48MX ;IF THESE TWO ARE EQUAL, WE WON'T BE PRINTING
2933 010254 001010 BNE 60$ ;ANY MORE LINES FOR A WHILE. SO,
2934 010256 PRINTX #FMT48I ; PUT OUT A MESSAGE TO THAT EFFECT.
010256 012746 010644 MOV #FMT48I,-(SP)
010262 012746 000001 MOV #1,-(SP)
010266 010600 MOV SP,R0
010270 104415 TRAP C$PNTX
010272 062706 000004 ADD #4,SP
2935 010276 60$:
2936 010276 000207 RTS PC
2937 .NLIST BEX
2938 010300 045 116 045 FMT48A: .ASCIZ \#N#S2#APRE OR (ALL VALUES IN OCTAL)\
2939 010347 045 123 065 FMT48B: .ASCIZ \#S5#AEXTENDED INFORMATION:\
2940 010402 045 116 045 FMT48C: .ASCIZ \#N#S3#APOST RAM SHOULD\
2941 010437 045 116 045 FMT48E: .ASCIZ \#N#S2#AWRITE ADDRESS BE IS XOR\
2942 010506 045 123 065 FMT48F: .ASCIZ \#S5#ABIT DATA SEQ LSB(DECIMAL)\
2943 010551 045 116 045 FMT48G: .ASCIZ \#N#S3#T#S4#04#S4#03#S3#03#S2#03\
2944 010611 045 123 066 FMT48H: .ASCIZ \#S6#01#S5#01#S3#T#S2#D2#A.\
2945 010644 045 116 045 FMT48I: .ASCIZ \#N#N#S5#AFURTHER DATA LINES SUPRESSED UNTIL NEW TEST DATA\
2946 010736 120 122 105 TXT48A: .ASCIZ \PRE \
2947 010743 120 117 123 TXT48B: .ASCIZ \POST\
2948 010750 040 106 127 TXT48C: .ASCIZ \FWD\
2949 010755 102 113 127 TXT48D: .ASCIZ \BKWD\
2950 .LIST BEX
2951 .EVEN
2952
2953
2954
2955 ;SBTTL ERROR HANDLER - ERR50 FOR REPORTING TIMER # 1 ERRORS
2956 ;
2957 010762 BGNMSG ERR50
010762 ERR50::
2958 010762 010146 MOV R1,(SP) ;SAVE R1 FOR CALLER
2959 010764 113701 002467 MOVB TMPB+1,R1 ;GET THE MODE LAST SETUP

```

ERROR HANDLER - ERR50 - FOR REPORTING TIMER # 1 ERRORS

```

2960 010770 000241 CLC ;SEEING AS THE CARRY BIT WILL BE ROTATED INTO
2961 ;THE DATA, WE HAD BETTER CLEAR IT JUST IN CASE.
2962 010772 042701 177477 BIC #C<BIT6*BIT7>,R1 ;LOOK @ JUST THE TIMER 1 MODE DEFINITION
2963 010776 106101 ROLB R1 ;POSITION IT FOR PRINTOUT
2964 011000 106101 ROLB R1
2965 011002 106101 ROLB R1
2966
2967 ;IDENTIFY THE MODE BEING USED AT THE TIME:
2968
2969 011004 PRINTX #FMT50A,R1
011004 010146 MOV R1,-(SP)
011006 012746 012747 MOV #FMT50A,(SP)
011012 012746 000002 MOV #2,-(SP)
011016 010600 MOV SP,R0
011020 104415 TRAP C$PNTX
011022 062706 000006 ADD #6,SP

2970 ;PRINT THE HEADING TO IDENTIFY THE REGISTERS:
2971
2972 011026 PRINTX #FMT50B
011026 012746 013021 MOV #FMT50B,(SP)
011032 012746 000001 MOV #1,(SP)
011036 010600 MOV SP,R0
011040 104415 TRAP C$PNTX
011042 062706 000004 ADD #4,SP

2973 ;AND THE VALUES THAT WERE LOADED INTO THE REGISTERS:
2974
2975 011046 PRINTX #FMT50C,#TXT8D,<B,TMP5*1>,<B,TMP4*1>,<B,TMP7*1>,<B,TMP6*1>
011046 005046 CLR (SP)
011050 153716 002455 BISB TMP6*1,(SP)
011054 005046 CLR -(SP)
011056 153716 002457 BISB TMP7*1,(SP)
011062 005046 CLR (SP)
011064 153716 002451 BISB TMP4*1,(SP)
011070 005046 CLR -(SP)
011072 153716 002453 BISB TMP5*1,(SP)
011076 012746 014053 MOV #TXT8D,-(SP)
011102 012746 013102 MOV #FMT50C,-(SP)
011106 012746 000006 MOV #6,-(SP)
011112 010600 MOV SP,R0
011114 104415 TRAP C$PNTX
011116 062706 000016 ADD #16,SP

2976 011122 PRINTX #FMT50D,<B,TMPB*1>,<B,TMPE*1>
011122 005046 CLR -(SP)
011124 153716 002475 BISB TMPE*1,(SP)
011130 005046 CLR -(SP)
011132 153716 002467 BISB TMPB*1,(SP)
011136 012746 013142 MOV #FMT50D,(SP)
011142 012746 000003 MOV #3,-(SP)
011146 010600 MOV SP,R0
011150 104415 TRAP C$PNTX
011152 062706 000010 ADD #10,SP

2977 ;AND THE VALUES READ FROM THOSE REGISTERS:
2978
2979 011156 PRINTX #FMT50E,#TXT8E,<B,TMP5>,<B,TMP4>,<B,TMP7>,<B,TMP6>
011156 005046 CLR (SP)
011160 153716 002454 BISB TMP6,(SP)
011164 005046 CLR (SP)

```

ERROR HANDLER ERR50 -- FOR REPORTING TIMER # 1 ERRORS

```

011166 153716 002456
011172 005046
011174 153716 002450
011200 005046
011202 153716 002452
011206 012746 014070
011212 012746 013102
011216 012746 000006
011222 010600
011224 104415
011226 062706 000016
2980 011232          PRINTX  #FMT50E,<B,TMPB>,<B,TMPD>
011232 005046
011234 153716 002472
011240 005046
011242 153716 002466
011246 012746 013157
011252 012746 000003
011256 010600
011260 104415
011262 062706 000010
2981
2982 011266 004737 012072      JSR      PC,NULERR      ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
2983 011272 012601          MOV      (SP)+,R1      ;RESTORE R1 FOR CALLER
2984 011274          ENDMSG
011274          L10015:
011274 104423          TRAP    C$MSG
2985
2986

```

```

BISB    TMP7,(SP)
CLR     -(SP)
BISB    TMP4,(SP)
CLR     (SP)
BISB    TMP5,(SP)
MOV     #TXT8E,(SP)
MOV     #FMT50C,(SP)
MOV     #6,(SP)
MOV     SP,R0
TRAP    C$PNTX
ADD     #16,SP

CLR     -(SP)
BISB    TMPD,(SP)
CLR     (SP)
BISB    TMPB,(SP)
MOV     #FMT50E,(SP)
MOV     #3,(SP)
MOV     SP,R0
TRAP    C$PNTX
ADD     #10,SP

```

ERROR HANDLER SUBROUTINES

```

2988 .SBTTL ERROR HANDLER SUBROUTINES
2989 ; -----
2990 ; SUBROUTINES USED ONLY BY ERROR HANDLERS -----
2991 ; -----
2992
2993 ; -----
2994 .SBTTL ERROR HANDLER SUBROUTINE - XORGB
2995 ; -----
2996 ; PERFORM EXCLUSIVE OR BETWEEN "GDATA" & "BDATA" PUTTING
2997 ; THE RESULT IN "XDATA"
2998
2999 XORGB: MOV R1,-(SP) ;PRESERVE WORKING REGISTER
3000 MOV GDATA,R1 ;GET "GOOD" DATA
3001 MOV BDATA,XDATA ;AND "BAD" DATA
3002 XOR R1,XDATA ;PERFORM EXCLUSIVE OR
3003 MOV (SP)+,R1 ;RESTORE R1
3004 RTS PC ;RETURN
3005
3006
3007 ; -----
3008 .SBTTL ERROR HANDLER SUBROUTINE - ERR4$
3009 ; -----
3010 ; IDENTIFY & DUMP THE BYTE SELECT REGISTERS
3011
3012 ERR4$: PRINTX @FMT4,@TXT3,@TXT1
3013 MOV @TXT1,(SP)
3014 MOV @TXT3,-(SP)
3015 MOV @FMT4,(SP)
3016 MOV @3,(SP)
3017 MOV SP,R0
3018 TRAP C$PNTX
3019 ADD @10,SP
3020
3021 PRINTX @FMT4A,<B,BSR0>,<B,BSR1>,<B,BSR2>,<B,BSR3>
3022 CLR (SP)
3023 BISB BSR3,(SP)
3024 CLR -(SP)
3025 BISB BSR2,(SP)
3026 CLR (SP)
3027 BISB BSR1,(SP)
3028 CLR -(SP)
3029 BISB BSR0,(SP)
3030 MOV @FMT4A,-(SP)
3031 MOV @5,-(SP)
3032 MOV SP,R0
3033 TRAP C$PNTX
3034 ADD @14,SP
3035
3036 PRINTX @FMT4B,@TXT2
3037 MOV @TXT2,(SP)
3038 MOV @FMT4B,(SP)
3039 MOV @2,(SP)
3040 MOV SP,R0
3041 TRAP C$PNTX
3042 ADD @6,SP
3043
3044 PRINTX @FMT4C,<B,BSR4>,<B,BSR5>,<B,BSR6>,<B,BSR7>
3045 CLR -(SP)
3046 BISB BSR7,(SP)
3047 CLR (SP)

```

ERROR HANDLER SUBROUTINE -- ERR4\$

```

011456 153716 002262
011462 005046
011464 153716 002260
011470 005046
011472 153716 002256
011476 012746 012345
011502 012746 000005
011506 010600
011510 104415
011512 062706 000014
3016 011516          PRINTX  #FMT4B,#TXT2A
011516 012746 013353
011522 012746 012340
011526 012746 000002
011532 010600
011534 104415
011536 062706 000006
3017 011542          PRINTX  #FMT4A,<B,BSR10>,<B,BSR11>,<B,BSR12>,<B,BSR13>
011542 005046
011544 153716 002274
011550 005046
011552 153716 002272
011556 005046
011560 153716 002270
011564 005046
011566 153716 002266
011572 012746 012305
011576 012746 000005
011602 010600
011604 104415
011606 062706 000014
3018 011612          PRINTX  #FMT4B,#TXT2B
011612 012746 013412
011616 012746 012340
011622 012746 000002
011626 010600
011630 104415
011632 062706 000006
3019 011636          PRINTX  #FMT4C,<B,BSR14>,<B,BSR15>,<B,BSR16>,<B,BSR17>
011636 005046
011640 153716 002304
011644 005046
011646 153716 002302
011652 005046
011654 153716 002300
011660 005046
011662 153716 002276
011666 012746 012345
011672 012746 000005
011676 010600
011700 104415
011702 062706 000014
3020 011706          RTS      PC
3021
3022
3023
3024

```

```

-----
:SBTTL .....ERROR HANDLER SUBROUTINE -- ERR5$
:
-----

```

.....ERROR HANDLER SUBROUTINE -- ERR5\$

```

3025          ; COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS
3026 011710   ERR5$:
3027 011710   PRINTX  #FMT4,#TXT6,#TXT4
              MOV     #TXT4,-(SP)
              MOV     #TXT6,(SP)
              MOV     #FMT4,-(SP)
              MOV     #3,-(SP)
              MOV     SP,RO
              TRAP   C$PNTX
              ADD     #10,SP
3028 011740   PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
              MOV     WSR6,-(SP)
              MOV     WSR4,-(SP)
              MOV     WSR2,-(SP)
              MOV     WSR0,-(SP)
              MOV     #FMT11,-(SP)
              MOV     #5,-(SP)
              MOV     SP,RO
              TRAP   C$PNTX
              ADD     #14,SP
3029 012000   PRINTX  #FMT4B,#TXT4A
              MOV     #TXT4A,-(SP)
              MOV     #FMT4B,-(SP)
              MOV     #2,-(SP)
              MOV     SP,RO
              TRAP   C$PNTX
              ADD     #6,SP
3030 012024   PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16
              MOV     WSR16,-(SP)
              MOV     WSR14,-(SP)
              MOV     WSR12,-(SP)
              MOV     WSR10,-(SP)
              MOV     #FMT11,-(SP)
              MOV     #5,-(SP)
              MOV     SP,RO
              TRAP   C$PNTX
              ADD     #14,SP
3031 012064   JSR     PC,NULERR          ;USE COMMON ROUTINE TO TERMINATE ERROR MESSAGE
3032 012070   RTS     PC
3033
3034          ;-----
3035          ;SBTTL          SUBROUTINE TO PERFORM "PRINTB  #ENDEMB"
3036
3037 012072   NULERR: PRINTB  #ENDEMB          ;TERMINATE ERROR MESSAGE
              MOV     #ENDEMB,-(SP)
              MOV     #1,(SP)
              MOV     SP,RO
              TRAP   C$PNTB
              ADD     #4,SP
3038 012112   RTS     PC
3039          ;-----

```

FORMAT SPEC'S FOR ERROR HANDLERS "FMT_"

```

3041 .SBTTL FORMAT SPEC'S FOR ERROR HANDLERS -- "FMT_"
3042 :-----
3043 :-----
3044 :-----
3045 :-----
3046 .NLIST BEX
3047 012114 045 116 045 ENDEMB: .ASCIZ /#N#N/
3048 012121 045 116 000 NEWLIN: .ASCIZ /#N/
3049 012124 045 116 045 FMT02: .ASCIZ /#N#AFAILING REG = #T#ASEL#02/
3050 012161 045 116 045 FMT02A: .ASCIZ /#N#A EXPECTED: #03#A ACTUAL: #03#A XOR: #03/
3051 012245 045 116 045 FMT4: .ASCIZ /#N#A THE CONTENTS OF ALL#T#N#T/
3052 012305 045 116 045 FMT4A: .ASCIZ /#N#S1#03#S5#03#S5#03#S5#03/
3053 012340 045 116 045 FMT4B: .ASCIZ /#N#T/
3054 012345 045 116 045 FMT4C: .ASCIZ /#N#S5#03#S5#03#S5#03#S5#03/
3055 012400 045 116 045 FMT5: .ASCIZ /#N#A WHEN #03#A LOADED INTO B# .1/
3056 012443 045 116 045 FMT5A: .ASCIZ /#N#A ATTEMPTING "M-LOOP" FUNCTION CODE #02#A ('#T#A)/
3057 012530 045 101 040 FMT07: .ASCIZ /#A DETECTED IN #T#T#A --/
3058 012562 045 116 045 FMT06: .ASCIZ /#N#N#T/
3059 012571 045 116 045 FMT06A: .ASCIZ /#N#T#03#S2#03#S2#03#S2#03#S2#03#S2#03/
3060 012637 045 123 062 FMT06B: .ASCIZ /#S2#03#S2#03/
3061 012654 045 116 045 FMT10: .ASCIZ /#N#A EXPECTED:#08#A ACTUAL:#08#A XOR:#08/
3062 012730 045 116 045 FMT11: .ASCIZ /#N#08#08#08#08/
3063 012747 045 116 045 FMT50A: .ASCIZ /#N#A TIMER # 1 MODE: #01#A REGISTERS:/
3064 013021 045 116 045 FMT50B: .ASCIZ /#N#S15#AT1CH T1CL T1LH T1LL ACR IFR IER/
3065 013102 045 116 045 FMT50C: .ASCIZ /#N#S3#T#S1#03#S3#03#S3#03#S3#03/
3066 013142 045 123 063 FMT50D: .ASCIZ /#S3#03#S9#03/
3067 013157 045 123 063 FMT50E: .ASCIZ /#S3#03#S3#03/
3068 013174 045 116 062 FMT50M: .ASCIZ /#N2#S10#A(T1CH & T1CL HAVEN'T YET BEEN LOADED)/
3069
3070 .SBTTL TEXT STRINGS FOR ERROR HANDLERS - "TXT_"
3071 :-----
3072 :-----
3073 :-----
3074 :-----
3075 013253 102 123 105 TXT1: .ASCIZ /BSEL0 BSEL1 BSEL2 BSEL3/
3076 013311 040 040 040 TXT2: .ASCIZ / BSEL4 BSEL5 BSEL6 BSEL7/
3077 013353 102 123 105 TXT2A: .ASCIZ /BSEL10 BSEL11 BSEL12 BSEL13/
3078 013412 040 040 040 TXT2B: .ASCIZ / BSEL14 BSEL15 BSEL16 BSEL17/
3079 013454 040 102 131 TXT3: .ASCIZ / BYTE SELECT REG'S ARE:/
3080 013504 040 040 040 TXT4: .ASCIZ / SEL0 SEL2 SEL4 SEL6/
3081 013544 040 040 040 TXT4A: .ASCIZ / SEL10 SEL12 SEL14 SEL16/
3082 013605 102 000 TXT5: .ASCIZ /B/
3083 013607 040 123 105 TXT6: .ASCIZ / SELECT REG'S ARE:/
3084 013632 040 122 105 TXT7: .ASCIZ / REGISTERS ORB ORA DDRB DDRA T1CL T1CH T1LL T1LH /
3085 013717 040 040 040 TXT7A: .ASCIZ / T2CL T2CH SR ACR PCR IFR IER ORA /
3086 014004 040 105 130 TXT8A: .ASCIZ / EXPECTED: /
3087 014021 040 101 103 TXT8B: .ASCIZ / ACTUAL: /
3088 014036 040 130 117 TXT8C: .ASCIZ / XOR: /
3089 014053 040 114 117 TXT8D: .ASCIZ / LOADED: /
3090 014070 040 122 105 TXT8E: .ASCIZ / READ: /
3091
3092 014105 116 117 120 TXTML0: .ASCIZ /NOP/
3093 014111 122 105 101 TXTML1: .ASCIZ /READ 1 BYTE/
3094 014125 127 122 111 TXTML2: .ASCIZ /WRITE 1 BYTE/
3095 014142 116 120 122 TXTML3: .ASCIZ /NPR-OUT 256 BYTES/
3096 014164 116 120 122 TXTML4: .ASCIZ /NPR-IN 256 BYTES/
3097 014205 123 105 124 TXTML5: .ASCIZ /SET MICROPROCESSOR S PC/

```

TEXT STRINGS FOR ERROR HANDLERS - "TXT..."

3098	014235	125	116	104	TXTML6:	.ASCIZ	/UNDEFINED/
3099	014247	123	105	124	TXTML7:	.ASCIZ	/SET MAINT INTR & CLR INTR DISABLE IN CPU STATUS/
3100							
3101	014327	126	111	101	TXTVR:	.ASCIZ	/VIA REGISTER /
3102	014345	117	122	102	TXTVR0:	.ASCIZ	/ORB/
3103	014351	117	122	101	TXTVR1:	.ASCIZ	/ORA/
3104	014355	104	104	122	TXTVR2:	.ASCIZ	/ODRB/
3105	014362	104	104	122	TXTVR3:	.ASCIZ	/DDRA/
3106	014367	124	061	103	TXTVR4:	.ASCIZ	/T1CL/
3107	014374	124	061	103	TXTVR5:	.ASCIZ	/T1CH/
3108	014401	124	061	114	TXTVR6:	.ASCIZ	/T1LL/
3109	014406	124	061	114	TXTVR7:	.ASCIZ	/T1LH/
3110	014413	124	062	103	TXTVR8:	.ASCIZ	/T2CL/
3111	014420	124	062	103	TXTVR9:	.ASCIZ	/T2CH/
3112	014425	123	122	000	TXTVRA:	.ASCIZ	/SR/
3113	014430	101	103	122	TXTVRB:	.ASCIZ	/ACR/
3114	014434	120	103	122	TXTVRC:	.ASCIZ	/PCR/
3115	014440	111	106	122	TXTVRD:	.ASCIZ	/IFR/
3116	014444	111	105	122	TXTVRE:	.ASCIZ	/IER/
3117	014450	117	122	101	TXTVRF:	.ASCIZ	/ORA/
3118							

ERROR MESSAGES "EM "

```

3120          .SBTTL  ERROR MESSAGES -- "EM_ "
3121          |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3122          |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3123          |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3124          |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3125 014454    115      111      103  EM3:   .ASCIZ  /MICRO-DIAG. FAILURE/
3126 014500    115      122      104  EM4:   .ASCIZ  /MRDY TIMEOUT/
3127 014515    115      101      123  EM5:   .ASCIZ  /MASTER CLR FAILURE/
3128 014540    103      123      122  EM6:   .ASCIZ  /CSR ADDRESS FAILURE/
3129 014564    103      123      122  EM7:   .ASCIZ  /CSR DATA PAT FAILURE/
3130 014611    102      123      105  EM8:   .ASCIZ  /BSELO SET=ALL ONES/
3131 014634    105      130      124  EM9:   .ASCIZ  /EXTERNAL BUS RESET FAILURE/
3132 014667    102      101      104  EM14:  .ASCIZ  /BAD CSR VALUE(S) AFTER MASTER CLEAR/
3133 014733    042      115      122  EM15:  .ASCIZ  /"MRDY" DIDN'T GO LOW WHILE PROCESSING A COMMAND/
3134 015013    104      115      126  EM16:  .ASCIZ  /DMV'S RAM LOC. (CORRESPONDING TO BSELO) NOT PROPERLY WRITTEN/
3135 015110    066      065      060  EM17:  .ASCIZ  /6502 WRITE FUNC. FAILURE AFTER "RUN" BIT IS SET/
3136 015170    066      065      060  EM17A: .ASCIZ  /6502 STILL RUNNING AFTER "RUN" BIT CLEARED/
3137 015243    126      111      101  EM20:  .ASCIZ  /VIA STATIC REGISTER ERROR/
3138 015275    126      111      101  EM20A: .ASCIZ  /VIA STATIC REGISTER ERROR -- TIMER NOT RUNNING/
3139 015355    126      111      101  EM20B: .ASCIZ  /VIA STATIC REGISTER ERROR -- TIMER CROSS TALK ERROR/
3140 015442    122      105      107  EM21:  .ASCIZ  /REGISTER NOT PROPERLY ZEROED/
3141 015477    132      105      122  EM22:  .ASCIZ  /ZEROING DDRB EFFECTED DDRA/
3142 015532    132      105      122  EM22A: .ASCIZ  /ZEROING DDRA EFFECTED DDRB/
3143 015565    122      105      101  EM25:  .ASCIZ  'READ/WRITE DATA ERROR'
3144 015613    125      116      105  EM34:  .ASCIZ  /UNEXPECTED "A" INTERRUPT/
3145 015644    125      116      105  EM34B: .ASCIZ  /UNEXPECTED "B" INTERRUPT/
3146 015675    122      101      115  EM47A: .ASCIZ  /RAM DATA ERROR ON INITIAL WRITE/
3147 015735    122      101      115  EM47B: .ASCIZ  /RAM DATA ERROR ON RE-READ AFTER TEST AREA FILLED/
3148 016016    122      101      115  EM48A: .ASCIZ  /RAM DATA ERROR -- MOVING INVERSIONS TEST/
3149 016067    042      124      061  EM50A: .ASCIZ  \ "T1" FLAG NOT CLEARED BY LOADING T1LH\
3150 016135    042      124      061  EM50B: .ASCIZ  \ "T1" FLAG NOT CLEARED BY LOADING T1CH\
3151 016203    042      124      061  EM50C: .ASCIZ  \ "T1" FLAG NOT CLEARED BY READING T1CL\
3152 016251    126      111      101  EM50D: .ASCIZ  \VIA'S T1CL NOT DECREMENTING\
3153 016305    126      111      101  EM50E: .ASCIZ  \VIA'S T1CH NOT DECREMENTING\
3154 016341    042      124      061  EM50F: .ASCIZ  \ "T1" FLAG NOT SET ON TIMER 1 TIMEOUT\
3155 016406    042      124      061  EM50G: .ASCIZ  \ "T1" FLAG CLEARED BY READING T1CH\
3156 016450    126      111      101  EM50H: .ASCIZ  \VIA'S T1LL IMPROPERLY LOADED BY WRITING T1CL @ ADDR 4\
3157 016536    042      124      061  EM50I: .ASCIZ  \ "T1" FLAG CLEARED BY READING T1LL\
3158 016600    126      111      101  EM50J: .ASCIZ  \VIA'S T1LH IMPROPERLY LOADED BY WRITING T1CH @ ADDR 5\
3159 016666    042      124      061  EM50K: .ASCIZ  \ "T1" FLAG CLEARED BY READING T1LH\
3160 016730    042      124      061  EM50L: .ASCIZ  \ "T1" FLAG NOT SET AFTER RE-LOADING T1CH & TIMEOUT\
3161 017012    042      124      061  EM50M: .ASCIZ  \ "T1" FLAG CLEARED BY LOADING T1LL\
3162 017054    042      124      061  EM50N: .ASCIZ  \ "T1" FLAG NOT CLEARED BY LOADING T1CH\
3163 017122    042      120      102  EM50S: .ASCIZ  \ "PB7" W/IN VIA NOT SET ON TIMER 1 TIMEOUT\
3164 017174    042      120      102  EM50U: .ASCIZ  \ "PB7" NOT SET AFTER TIMER 1 TIMEOUT\
3165 017240    042      120      102  EM50V: .ASCIZ  \ "PB7" NOT DRIVEN LOW BY LOADING T1CH\
3166 017305    042      120      102  EM50W: .ASCIZ  \ "PB7" UNEXPECTEDLY MODIFIED BY TIMER 1\
3167 017354    042      124      061  EM50X: .ASCIZ  \ "T1" NOT RESET AFTER BEING CLEARED\
3168 017417    042      120      102  EM50Y: .ASCIZ  \ "PC7" PREMATURELY SET DURING T1 COUNTDOWN\
3169 017471    042      120      102  EM50Z: .ASCIZ  \ "PB7" NOT SET AFTER SECOND CYCLE\
3170
3171          .EVEN
    
```

TEXT ADDRESS TABLES FOR ERROR HANDLERS "TXT T"

```

3173 .SBTTL TEXT ADDRESS TABLES FOR ERROR HANDLERS "TXT T"
3174 ;
3175 ; - - - - - TEXT ADDRESS TABLES USED BY ERROR HANDLERS - - - - -
3176 ; - - - - -
3177
3178 017532 014105 014111 014125 TXTMLT: .WORD TXTML0, TXTML1, TXTML2, TXTML3, TXTML4, TXTML5, TXTML6, TXTML7
3179
3180 017552 014327 .WORD TXTVR
3181 017554 014345 014351 014355 TXTVRT: .WORD TXTVR0, TXTVR1, TXTVR2, TXTVR3, TXTVR4, TXTVR5, TXTVR6, TXTVR7
3182 017574 014413 014420 014425 .WORD TXTVR8, TXTVR9, TXTVRA, TXTVRB, TXTVRC, TXTVRD, TXTVRE, TXTVRF
3183
3184 .LIST BEX

```

LOAD DEVICE PROTECTION TABLE

.SBTTL LOAD DEVICE PROTECTION TABLE

```

;////////////////////////////////////
;/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
;/ PROTECTED FROM TESTING. IF DESIRED.
;////////////////////////////////////

```

```

3186
3187
3188
3189
3190
3191
3192
3193 017614
      017614
3194 017614 177777
3195 017616 177777
3196 017620 177777
3197 017622

```

BGNPROT

```

.WORD -1 ;DON'T CHK CSR ADRS
.WORD 1 ;DON'T CHK MASSBUS UNIT NO.
.WORD 1 ;DON'T CHK DRIVE NO.
ENDPROT

```

L\$PROT::

INITIALIZE SECTION

```

3199          .SBTTL INITIALIZE SECTION
3200
3201          ;////////////////////////////////////
3202          ;// THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3203          ;// AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
3204          ;////////////////////////////////////
3205
3206 017622          BGNINIT
3207          017622          L$INIT::
3208 017622          SFTVEC #140,#170000,#340          ;ODT ROM ADDRESS
3209          017622 012746 000340          MOV          ;JB REV A-0
3210          017626 012746 170000          MOV          #340,-(SP)
3211          017632 012746 000140          MOV          #170000,(SP)
3212          017636 012746 000003          MOV          #140,-(SP)
3213          017642 104437          MOV          #3,-(SP)
3214          017644 062706 000010          TRAP         C$SVEC
3215          017650 010637 002324          ADD          #10,SP
3216          017654          ;SEE IF PROGRAM JUST STARTED, BR IF YES
3217          017654 012700 000040          MOV          SP,PSTACK          ;SAVE BASE-LEVEL STACK POINTER
3218          017660 104447          READEF #EF.START
3219          017662          BCOMPLETE          STARST          MOV          #EF.START,R0
3220          017662 103417          TRAP         C$REFG
3221          017664          ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
3222          017664 012700 000037          BCS          STARST
3223          017670 104447          READEF #EF.RESTART
3224          017672          BCOMPLETE          RESTRT          MOV          #EF.RESTART,R0
3225          017672 103435          TRAP         C$REFG
3226          017674          ;SEE IF THIS IS A NEW PASS, BR IF YES
3227          017674 012700 000035          BCS          RESTRT
3228          017700 104447          READEF #EF.NEW
3229          017702          BCOMPLETE          NEWST          MOV          #EF.NEW,R0
3230          017702 103433          TRAP         C$REFG
3231          017704          ;SEE IF PROGRAM WAS JUST CONTINUED
3232          017704 012700 000036          BCS          NEWST
3233          017710 104447          READEF #EF.CONTINUE
3234          017712          BCOMPLETE          10$          MOV          #EF.CONTINUE,R0
3235          017712 103401          TRAP         C$REFG
3236          017714 000436          BR          GETPRM          BCS          10$
3237          017716 000137 020122          10$: JMP          CONTIN          ;(THIS IS TO FAR AWAY FOR A "BR" INSTRUCTION)
3238          017722          STARST:          ;ENTER HERE IF "START" COMMAND ISSUED
3239          017722          ; TEST FOR THE PRESENCE OR ABSENCE OF A CONSOLE TERMINAL.
3240          017722 005037 002346          CLR          CONSOL          ;RESET THE CONSOLE TERMINAL FLAG
3241          017726          SETVEC #4,#CONST,#0          ;SETUP BUS TIMEOUT VECTOR TO TEST FOR A CONSOLE
3242          017726 012746 000000          MOV          #0,(SP)
3243          017732 012746 020222          MOV          #CONST,-(SP)
3244          017736 012746 000004          MOV          #4,(SP)
3245          017742 012746 000003          MOV          #3,(SP)

```

INITIALIZE SECTION

```

017746 104437
017750 062706 000010
3233 017754 005737 177564          TST      #0177564      ;TRY TO ACCESS THE CONSOLE TERMINAL S "XCSR"
3234 017760          CLRVEC  #4              ;WE SHOULD BE THROUGH WITH THIS BY NOW
017760 012700 000004          MOV      #4,R0
017764 104436          TRAP    C$CVEC
3235
3236 017766          RESTRT:      ;ENTER HERE IF "RESTART" COMMAND ISSUED
3237
3238          ;CLEAR DEVICE MAP
3239 017766 005037 002342          CLR      DEVMAP
3240
3241 017772          NEWST:      ;ENTER HERE BEFORE EACH TEST
3242
3243 017772 012737 177777 002322          MOV      #-1,LOGDEV      ;RESFT LOGICAL DEVICE TO -1
3244 020000 005237 002340          INC      FRSPAS          ;INCREMENT NO. OF PASSES AFTER LOAD
3245 020004 012737 000001 002344          MOV      #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
3246          ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
3247          ; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
3248 020012          GETPRM:
3249 020012 005237 002322          INC      LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
3250 020016          GPHARD LOGDEV,R1        ;GET P-TABLE POINTER INTO R1
020016 013700 002322          MOV      LOGDEV,R0
020022 104442          TRAP    C$GPHRD
020024 010001          MOV      R0,R1
3251 020026          BCOMPLETE 10$          ;BR IF DEVICE AVAILABLE
020026 103403          BCS     10$
3252 020030 006337 002344          ASL      DEVPTR          ;IF UN-AVAILABLE, SHIFT DEVICE MAP BIT POINTER
3253 020034 000766          BR       GETPRM         ; AND SKIP THIS DEVICE
3254
3255 020036 053737 002344 002342 10$:  BIS      DEVPTR,DEVMAP    ;ELSE, SET BIT FOR THIS DEVICE IN DEVICE MAP
3256 020044 006337 002344          ASL      DEVPTR          ;SHIFT DEVICE MAP BIT POINTER
3257
3258          ; "R1" WAS RETURNED WITH A POINTER TO THE CURRENT "P TABLE"
3259
3260 020050 012100          MOV      (R1),R0         ;GET THE DEVICE CSR ADDRESS
3261 020052 012703 000020          MOV      #16,R3          ;WE HAVE TO SETUP THIS MANY ADDRESS POINTERS
3262 020056 012702 002352          MOV      #MPCSR,R2       ;THIS IS THE ADDRESS OF THE FIRST POINTER
3263 020062 010022 12$:  MOV      R0,(R2)+         ;SETUP ONE CSR POINTER
3264 020064 005200          INC      R0              ;POINT TO THE NEXT CSR ADDRESS
3265 020066 077303          SOB     R3,12$          ;LOOP AS LONG AS THERE ARE MORE TABLE ENTRIES
3266          ;ELSE, FALL THROUGH TO CONTINUE GETTING MORE
3267          ; P TABLE DATA
3268
3269 020070 012100          MOV      (R1),R0         ;GET INTERRUPT VECTOR
3270 020072 010037 002412          MOV      R0,MPIVEC       ;SETUP "A" VECTOR POINTER
3271 020076 022020          CMP     (R0),(R0)+       ;ADD 4 TO VECTOR TO GET ADDRESS OF 'B' VECTOR
3272 020100 010037 002414          MOV     R0,MPOVEC        ;SETUP "B" VECTOR POINTER
3273
3274 020104 012100          MOV      (R1),R0         ;GET DMV11 DEVICE PRIORITY
3275 020106 006200          ASR     R0              ; RE POSITION IT
3276 020110 006200          ASR     R0
3277 020112 006200          ASR     R0
3278 020114 006200          ASR     R0
3279 020116 010037 002416          MOV     R0,MPRIOR        ;SETUP OUR VARIABLE FOR INT. VECTOR INIT S
3280
3281 020122          CONTIN:      ;ENTER HERE WHEN A "CONTINUE" COMMAND IS ISSUED

```

INITIALIZE SECTION

```

3282
3283 020122          SETVEC @@MPIVEC,@MPIHAN,@@MPRIOR ;SETUP "A" INT. VECTOR
      020122 013746 002416          MOV      @@MPRIOR,(SP)
      020126 012746 005152          MOV      @MPIHAN,-(SP)
      020132 013746 002412          MOV      @@MPIVEC,(SP)
      020136 012746 000003          MOV      @3,-(SP)
      020142 104437          TRAP     C$SVEC
      020144 062706 000010          ADD      @10,SP
3284 020150 005037 005222          CLR      IHILNK          ;WE DON'T WANT THE HANDLER TO LINK ELSEWHERE
3285 020154          SETVEC @@MPOVEC,@MPOHAN,@@MPRIOR ;SETUP "B" INT. VECTOR
      020154 013746 002416          MOV      @@MPRIOR,(SP)
      020160 012746 005224          MOV      @MPOHAN,(SP)
      020164 013746 002414          MOV      @@MPOVEC,(SP)
      020170 012746 000003          MOV      @3,(SP)
      020174 104437          TRAP     C$SVEC
      020176 062706 000010          ADD      @10,SP
3286 020202 005037 005274          CLR      IHOLNK          ;WE DON'T WANT THE HANDLER TO LINK ELSEWHERE
3287 020206 005037 002330          CLR      INTWCH        ;RESET "INTERRUPT WATCH" FLAGS (BOTH "A" & "B")
3288
3289 020212 012737 000001 002336          MOV      @1,FRSTIM      ;MARK FLAG FOR NEXT TIME THROUGH
3290 020220          ENDINIT          ;END OF "INIT" CODE
      020220          L10017:
      020220 104411          TRAP     C$INIT
3291
3292          ; ***** SUBROUTINES USED BY "INIT" CODE *****
3293
3294          ; INTERRUPT HANDLER FOR CONSOLE TERMINAL PRESENCE TESTING
3295
3296 020222 012737 177777 002346          CONST: MOV      @-1,CONSOL      ;INDICATE THAT NO CONSOLE TERMINAL EXISTS!
3297 020230 000002          RTI          ;RETURN
3298

```

AUTO DROP UNIT SECTION

```

3300 .SBTTL AUTO DROP UNIT SECTION
3301
3302 ;////////////////////////////////////
3303 ;// THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
3304 ;// WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
3305 ;////////////////////////////////////
3306
3307 ;*****
3308 ;
3309 ; THIS ALGORITHM IS THE SAME AS TEST # 1 EXCEPT THAT TEST 1
3310 ; WILL JUST REPORT THE FAILURE AND GO ON THIS ROUTINE WILL CAUSE THE
3311 ; DEVICE TO BE DROPPED IF A BUS-TIMEOUT OCCURS WHEN ANY OF THE CSR'S
3312 ; ARE ACCESSED WITH EITHER A "TST" OR "TSTB" INSTRUCTION.
3313 ;
3314 ;-----*****
3315
3316 020232          BGNAUTO
3317 020232          SETVEC  #4,#AD.HIT,#0 ;SETUP INVALID ADDRESS TRAP VECTOR
3318 020232 012746 000000          MOV      #0,(SP)
3319 020236 012746 020350          MOV      #AD.HIT,-(SP)
3320 020242 012746 000004          MOV      #4,(SP)
3321 020246 012746 000003          MOV      #3,-(SP)
3322 020252 104437          TRAP    C$SVEC
3323 020254 062706 000010          ADD      #10,SP
3324 020260 005037 002440          CLR      TMPO ;INITIALIZE TRAP FLAG REGISTER
3325 020264 012702 000001          MOV      #1,R2 ;FLAG BIT
3326 020270 013703 002352          MOV      BSELO,R3 ;INIT ADDRESS POINTER
3327 020274 105723          1$: TSTB   (R3)+ ;ACCESS THE CSR S BY BYTES.
3328 020276 006302          ASL      R2
3329 020300 103375          BCC      1$
3330 020302 013703 002352          MOV      BSELO,R3 ;RE INIT ADDRESS POINTER
3331 020306 012702 000001          MOV      #1,R2 ;RE INIT FLAG BIT
3332 020312 005723          2$: TST   (R3)+ ;ACCESS THE CSR S BY WORDS.
3333 020314 006302          ASL      R2
3334 020316 006302          ASL      R2
3335 020320 103374          BCC      2$
3336 020322          CLRVEC  #4 ;RESTORE THE VECTOR TO DS
3337 020322 012700 000004          MOV      #4,R0
3338 020326 104436          TRAP    C$CVEC
3339 020330 005737 002440          TST      TMPO ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
3340 020334 001403          BEQ      AD.OK ;NO, EXIT TEST
3341 020336          DODU    LOGDEV ;YES, DROP THIS LOGICAL DEV.
3342 020336 013700 002322          MOV      LOGDEV,R0
3343 020342 104451          TRAP    C$DODU
3344 020344 000240          AD.OK: NOP ;(FOR PATCHING IN A HALT IF NECESSARY)
3345 020346          ENDAUTO
3346 020346          L10020:
3347 020346 104461          TRAP    C$AUTO
3348 020350 050237 002440          AD.HIT: BIS   R2, TMPO ;FLAG THE HIT IF WE GET IT!
3349 020354 000002          RTI ;RETURN
    
```

CLEANUP CODING SECTION

.SBTTL CLEANUP CODING SECTION

```

;////////////////////////////////////
; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
; AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
;////////////////////////////////////

```

```

3342
3343
3344
3345
3346
3347
3348
3349 020356          BGNCLN
      020356
3350 020356          CLRVEC @MPIVEC          ;RETURN VECTORS TO SUPERVISOR
      020356 013700 002412
      020362 104436
      L$CLEAN::
      MOV @MPIVEC,RO
      TRAP C$CVEC
3351 020364          CLRVEC @MPOVEC
      020364 013700 002414
      020370 104436
      MOV @MPOVEC,RO
      TRAP C$CVEC
3352 020372          ENDCLN
      020372
      L10021:
      020372 104412
      TRAP C$CLEAN

```


DROP UNIT SECTION

3354
 3355
 3356
 3357
 3358
 3359
 3360
 3361 020374
 020374
 3362
 3363 020374
 020374 104433
 3364 020376
 020376 104453

.SBTTL DROP UNIT SECTION

;/;;;
 ;/ THE DROP UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 ;/ TO NO LONGER BE TESTED.
 ;/;;;

BGNDU

;ISSUE UNIBUS RESET TO CLEAN UP
 BRESET

ENDDU

L\$DU::

TRAP C\$RESET

L10022:

TRAP C\$DU

ADD UNIT SECTION

3366
 3367
 3368
 3369
 3370
 3371
 3372
 3373
 3374 020400
 020400
 3375 020400
 020400
 020400 104452

.SBTTL ADD UNIT SECTION

```

:////////////////////
:/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:/ TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
:/ "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
:////////////////////

```

BGNAU

ENDAU

L\$AU::

L10023:

TRAP C\$AU

TEST 1 DMV 11 AVAILABILITY

3393

.SBTTL TEST 1 - DMV-11 AVAILABILITY

```

*****
;*
;* TEST 1 -- DMV-11 AVAILABILITY
;*
;* EACH NORMALLY USED CSR IS ACCESSED WITH A "TST" OR "TSTB" INSTRUCTION AND IF
;* A BUS TIMEOUT OCCURS (INTERRUPT @ VECTOR ADDR 4) A FLAG WILL BE SET SHOWING
;* WHICH CSR ADDR AND INSTRUCTION FAILED. "T1.HSW" REFLECTS "TST" INSTRUCTIONS
;* AND "T1.HSB" REFLECTS "TSTB" INSTRUCTIONS.
;*
;* EXAMPLES:
;*
;* IF "TSTB @BSEL1" FAILS, BIT # 1 OF "T1.HSB" WILL BE SET.
;* IF "TST @SEL4" FAILS, BIT # 4 OF "T1.HSW" WILL BE SET
;* (NOTE: ONLY EVEN BITS IN "T1.HSW" CAN BE SET).
;*
;* THE FLAG WORDS ARE OUTPUT IN BINARY AS "EXTENDED ERROR INFORMATION".
;*****

```

```

;
; BGNTST
;
; T1::
3394 020402 005037 020546 CLR T1.HSW ;INITIALIZE TRAP FLAG REGISTER
3395 020406 012702 000001 MOV #1,R2 ;FLAG BIT FOR BYTE ACCESSED CSR 0.
3396 020412 013703 002352 MOV BSEL0,R3 ;INIT ADDRESS POINTER
3397 020416 012746 000000 SETVEC #4,#T1.HIT,#0 ;SETUP INVALID-ADDRESS TRAP VECTOR
;
; MOV #0,-(SP)
; MOV #T1.HIT,-(SP)
; MOV #4,-(SP)
; MOV #3,-(SP)
; TRAP C$SVEC
; ADD #10,SP
3398
3399 020444 105723 1$: TSTB (R3)+ ;ACCESS THE CSR'S BY BYTES.
3400 020446 006302 ASL R2
3401 020450 103375 BCC 1$
3402
3403 020452 013737 020546 020550 MOV T1.HSW,T1.HSB ;MOVE BYTE INTERRUPT FLAG TO PROPER LOCATION.
3404 020460 005037 020546 CLR T1.HSW ;INITIALIZE TRAP FLAG REGISTER
3405 020464 012702 000001 MOV #1,R2 ;FLAG BIT FOR WORD ACCESSED CSR 0.
3406 020470 013703 002352 MOV BSEL0,R3 ;RE-INIT ADDRESS POINTER
3407
3408 020474 005723 2$: TST (R3)+ ;ACCESS THE CSR'S BY WORDS.
3409 020476 006302 ASL R2
3410 020500 006302 ASL R2
3411 020502 103374 BCC 2$
3412
3413 020504 CLRVEC #4 ;RESTORE THE VECTOR TO DS
; MOV #4,R0
; TRAP C$CVEC
3414 020512 005737 020546 TST T1.HSW ;DID WE GET AN INVALID ADDRESS TRAP?
3415 020516 001003 BNE 3$ ;YES, REPORT FAILURE
3416 020520 005737 020550 TST T1.HSB
3417 020524 001404 BEQ T1.OK
3418 020526 3$: GEOD T1.EMD,T1.EM1 ;YES, REPORT THE ERROR
; "DEVICE FATAL" ERROR # 9

```

TEST 1 -- DMV-11 AVAILABILITY

```

020526 104455
020530 000011
020532 020670
020534 020552
3419 020536 T1.OK: ENDTST
020536
020536 104401
3420
3421 020540 050237 020546 .1.HIT: BIS R2,T1.HSW ;FLAG THE HIT IF WE GET IT!
3422 020544 000002 RTI ;RETURN
3423
3424 020546 000000 T1.HSW: .WORD 0 ;INVALID ADDRESS TRAP FLAG WORD:
3425 ;BITS SET INDICATE TRAPS ON WORD ACCESSES
3426 ;(BIT # SET = CSR # THAT FAILED)
3427 020550 000000 T1.HSB: .WORD 0 ;INVALID ADDRESS TRAP FLAG WORD:
3428 ;BITS SET INDICATE TRAPS ON BYTE ACCESSES
3429 ;(BIT # SET = CSR # THAT FAILED)
3430 020552 BGNMSG T1.EM1
020552
3431 020552 PRINTB #T1.1,MPCSR ;IDENTIFY ERROR AND ON WHAT DEVICE
020552 013746 002352 MOV MPCSR,-(SP)
020556 012746 020717 MOV #T1.1,-(SP)
020562 012746 000002 MOV #2,-(SP)
020566 010600 MOV SP,RO
020570 104414 TRAP C#PNTB
020572 062706 000006 ADD #6,SP
3432 020576 PRINTX #T1.2 ;IF REQUESTED, ALSO INDICATE MISSES (TRAPS)
020576 012746 021001 MOV #T1.2,-(SP)
020602 012746 000001 MOV #1,-(SP)
020606 010600 MOV SP,RO
020610 104415 TRAP C#PNTX
020612 062706 000004 ADD #4,SP
3433 020616 PRINTX #T1.3
020616 012746 021034 MOV #T1.3,-(SP)
020622 012746 000001 MOV #1,-(SP)
020626 010600 MOV SP,RO
020630 104415 TRAP C#PNTX
020632 062706 000004 ADD #4,SP
3434 020636 PRINTX #T1.4,T1.HSW,T1.HSB
020636 013746 020550 MOV T1.HSB,(SP)
020642 013746 020546 MOV T1.HSW,-(SP)
020646 012746 021106 MOV #T1.4,-(SP)
020652 012746 000003 MOV #3,-(SP)
020656 010600 MOV SP,RO
020660 104415 TRAP C#PNTX
020662 062706 000010 ADD #10,SP
3435 020666 ENDMSG
020666
020666 104423
3436
3437 .NLIST BEX
3438 020670 101 126 101 T1.EMD: .ASCIZ 'AVAILABILITY TEST (#1)'
3439 020717 045 116 045 T1.1: .ASCIZ '#N#ADMV-11 @ #O#A NOT RESPONDING TO CSR ACCESSING'
3440 021001 045 116 062 T1.2: .ASCIZ '#N2#S21#ASEL #S11#ABSEL #'
3441 021034 045 116 045 T1.3: .ASCIZ '#N#S15#AE C A 8 6 4 2 0 FEDCBA9876543210'
3442 021106 045 116 062 T1.4: .ASCIZ '#N2#A TRAP FLAGS:#B16#S2#B16'
3443 .LIST BEX

```

L10024:

TRAP C#ERDF
.WORD 9
.WORD T1.EHD
.WORD T1.EM1

TRAP C#ETST

T1.EM1::

MOV MPCSR,-(SP)
MOV #T1.1,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C#PNTB
ADD #6,SP
MOV #T1.2,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C#PNTX
ADD #4,SP
MOV #T1.3,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C#PNTX
ADD #4,SP
MOV T1.HSB,(SP)
MOV T1.HSW,-(SP)
MOV #T1.4,-(SP)
MOV #3,-(SP)
MOV SP,RO
TRAP C#PNTX
ADD #10,SP

L10025:

TRAP C#MSG

B8

TEST 1 DMV 11 AVAILABILITY

SEQ 0092

3444

.EVEN

3461

.SBTTL TEST 2 - MASTER CLEAR, RUN MICRODIAGNOSTICS

.....

TEST 2 - MASTER CLEAR, RUN MICRODIAGNOSTICS

A MASTER CLEAR IS ISSUED TO THE DMV-11, AND THE PROGRAM ALLOWS SUFFICIENT TIME FOR THE MICRODIAGNOSTICS TO BE PERFORMED. THESE MICRODIAGNOSTICS RESIDE IN 6502 PROGRAM MEMORY, AND THOROUGHLY VERIFY THE OPERATION OF THE 6502 MICROPROCESSOR CHIP. THEN, THEY CHECK OUT THE DATA RAM, THE 6502'S ACCESS TO THE CSR'S, AND PERFORM A SIMPLE MESSAGE TEST USING THE 6522 CHIP AND THE USYRT, WITH INTERNAL LOOPBACK.

NEXT, THE LSI-11 PROGRAM READS THE THE CSR'S (SEL0-SEL6) AND CHECKS THEM FOR THEIR EXPECTED INITIALIZED STATES. IF AN ERROR HAS OCCURRED IN THE MICRO DIAGNOSTICS THE NUMBER OF THE FAILING TEST WILL BE FOUND IN SEL4, AND RUN (BIT 7) WILL NOT BE SET IN BSEL1.

.....

BGNTST

ISSUE A MASTER CLEAR, AND DELAY FOR MICRO-DIAGNOSTICS TO COMPLETE BY CALLING SUBROUTINE MASCLR. T2:

021146
 3462
 3463
 3464
 3465 021146 004737 003614
 3466 021152 103002
 3467 021154
 021154 104460
 3468 021156 000436
 3469
 3470
 3471 021160 005001
 3472 021162 005002
 3473 021164 016203 003040
 3474 021170 062702 000002
 3475 021174 126271 003040 002352
 3476 021202 001005
 3477 021204 005202
 3478 021206 005201
 3479 021210 005201
 3480
 3481 021212 077310
 3482
 3483 021214 000417
 3484
 3485 021216 117137 002352 002312 1:
 3486 021224 004737 004534
 3487 021230 116237 003040 002310
 3488 021236 006201
 3489 021240 010137 002334
 3490 021244
 021244 104455
 021246 000017
 021250 014667
 021252 005304

```

JSR    PC,MASCLR    ; ATTEMPT- TO RUN THE MICRO DIAGNOSTIC
BCC    B#           ; IF NO ERROR, PROCEED
ERROR  ; ELSE, REPORT IT AND
                                TRAP    C$ERROR
BR     24#         ; EXIT THIS TEST

; FIRST, INITIALIZE INDEX REGISTERS
B#:   CLR    R1    ; R1 IS THE INDEX OF THE BYTE SELECT TABLE
      CLR    R2    ; R2 IS THE INDEX OF THE RESULTS TABLE
      MOV    RESFMC(R2),R3 ; GET THE NUMBER OF PATTERNS IN RESULTS TABLE
      ADD    #2,R2 ; MOVE POINTER TO NEXT BYTE
2#:   CMPB   RESFMC(R2),BSEL(R1) ; COMPARE EXPECTED RESULTS WITH CSR'S.
      BNE    1#    ; A MISMATCH IS A DEVICE FATAL ERROR
      INC    R2    ; INCREMENT TABLE POINTER
      INC    R1    ; INCREMENT POINTER
      INC    R1    ; BY 2 (WORD INCREMENT)
SOB   R3,2#       ; CONTINUE TO LOOP THROUGH TABLE
BR    24#         ; TEST COMPLETE WITH NO ERRORS, GO END TEST.
1#:   MOVB   BSEL(R1),BDATA ; GET DATA BYTE THAT FAILED
      JSR    PC,GETBSR    ; GET THE BSEL REGISTERS FOR DUMPING
      MOVB   RESFMC(R2),GDATA ; GET EXPECTED RESULT FROM TABLE
      ASR    R1    ; CONVERT WORD OFFSET TO BYTE CSR #
      MOV    R1,REGNUM
      GEDF   EM14,ERR2    ; DEVICE FATAL ERROR, REPORT IT AND END TEST
                                ; "DEVICE FATAL" ERROR # 10
                                TRAP    C$ERDF
                                .WORD  10
                                .WORD  EM14
                                .WORD  ERR2
    
```

D8

CNDMA40 DMV11 MCTRL DIAG #1 MACRO M1200 22 FEB 84 15:22 PAGE 52 1

TEST 2 MASTER CLEAR, RUN MICRODIAGNOSTICS

SEQ 0094

3491 021254 244: ENDTST
 021254
 021254 104401

L10026: TRAP C#ETST

TEST 3 CSR ADDRESSING

3503

.SBTTL TEST 3 CSR ADDRESSING

```

;*****
;*
;* TEST 3 - CSR ADDRESSING
;*
;* FIRST, HALT THE 6502 UP BY CLEARING ALL CSRS. THEN, WRITE A DIFFERENT WORD
;* OF DATA PATTERN A INTO EACH OF BSEL0 17, AND AFTER EACH WRITE, READ AND
;* COMPARE ALL REGS TO EXPECTED VALUES.
;*
;* DATA PATTERN A = 001, 002, 004, 010, 020, 040, 100, 200, 052, 300, 140,
;* 060, 030, 014, 006, 003
;*
;*****

```

; BGNTST

T3::

```

;-----
; ***** DETAILED TEST DESCRIPTION *****
; THIS TEST PROCEEDS AS FOLLOWS:
;
; (1) CLEAR ALL CSRS AND VERIFY SAME (CLEARING BSEL01 HALTS 6502)
; (2) WRITE 01 INTO BSEL0; VERIFY BSEL0=01,ALL OTHERS=0
; (3) WRITE 02 INTO BSEL1; VERIFY BSEL0=01,BSEL1=02,ALL OTHERS=0.
; (4) WRITE 04 INTO BSEL2; VERIFY BSEL0=01,BSEL1=02,BSEL2=04,ALL OTHERS=0
;
; (5) -> (17) CONTINUE TO INCREMENTALLY WRITE DATA-PATTERN-A INTO THE BSR'S.
; CHECKING ALL BSR S AFTER EACH WRITE, UNTIL BSR'S COMPLETELY
; FILLED WITH DATA-PATTERN-A.
; NOTE: IF AN ERROR OCCURS, THE FIRST BAD BSR NUMBER AND GOOD/BAD VALUES ARE
; GIVEN, FOLLOWED BY A COMPLETE BSR DUMP.
;-----

```

; CLEAR DMV CSRS AND RESULTS TABLE

```

3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522 021256 012703 000010
3523 021262 013701 002352
3524 021266 012702 003062
3525 021272 005021
3526 021274 005022
3527 021276 005022
3528 021300 077304
3529
3530
3531
3532 021302 005002
3533 021304 012703 000020
3534 021310 105772 002352
3535 021314 001035
3536 021316 005722
3537 021320 077305
3538
3539
3540
3541
3542 021322 005001
3543 021324 012703 000020

```

```

MOV #10,R3 ;GET # OF CSRS
MOV #BSEL,R1 ;GET 1ST CSR ADDRESS
MOV #RESFT3,R2 ;GET 1ST RESULTS TABLE ADDRESS
1$: CLR (R1)+ ;CLEAR CSR, BUMP POINTER
CLR (R2)+ ;CLEAR RESULTS TABLE LOC., BUMP POINTER
CLR (R2)+ ; AND DO AGAIN
SOB R3,1$ ;LOOP UNTIL ALL DONE

```

; NOW VERIFY CSRS ARE ALL ZEROED

```

CLR R2 ;CLEAR BSR ADDRESS INDEX
MOV #CSREGS,R3 ;GET # OF CSRS
2$: TSTB #BSEL(R2) ;IS THIS CSR=0 ?
BNE 5$ ;IF NO: GO REPORT ERROR
TST (R2)+ ; YES: BUMP INDEX
SOB R3,2$ ;DO UNTIL ALL BSRS CHECKED

```

; INITIALIZE INDEX REGISTERS

```

CLR R1 ;INITIALIZE PATTERN INDEX REGISTER
MOV #CSREGS,R3 ;GET NUMBER OF CSR'S

```


TEST 3 CSR ADDRESSING

```

3544
3545 ; THE FIRST WORD OF THE DATA TABLE CONTAINS THE NUMBER OF PATTERNS IN
3546 ; THE TABLE:
3547 021330 016104 002504 MOV PATA(R1),R4 ;INITIALIZE NUMBER OF PATTERNS COUNT
3548 021334 005721 TST (R1)+ ;MOVE TABLE POINTER
3549
3550 ; PUT NEXT PATTERN OF DATA INTO NEXT REGISTER AND TEST AREA:
3551 ;
3552 ; CALCULATE INDEX INTO DATA AREA AND TO REGISTER
3553
3554 021336 010102 3$: MOV R1,R2 ;GET INDEX INTO TEST DATA AREA
3555 021340 005742 TST (R2) ;IT'S ONE WORD TOO LARGE
3556 021342 006302 ASL R2 ;CONVERT FROM BYTE TO WORD INDEX
3557
3558 ; NOW, SETUP THE EXPECTED RESULTS AREA AND LOAD THE SELECT REGISTER
3559
3560 021344 116162 002504 003062 MOVB PATA(R1),RESFT3(R2) ;UPDATE THE EXPECTED RESULTS TABLE
3561 021352 116172 002504 002352 MOVB PATA(R1),@BSEL(R2) ;PUT PATTERN INTO THE CSR
3562
3563 021360 005201 INC R1 ;BUMP DATA POINTER FOR NEXT TIME AROUND
3564 021362 005002 CLR R2 ;INITIALIZE TABLE INDEX
3565 021364 012703 000020 MOV @CSREGS,R3 ;INITIALIZE NUMBER OF REGISTERS
3566
3567 021370 126272 003062 002352 4$: CMPB RESFT3(R2),@BSEL(R2) ;COMPARE CSR WITH RESULTS TABLE
3568 021376 001004 BNE 5$ ;A MISMATCH IS A DEVICE FATAL ERROR
3569 021400 005722 TST (R2)+ ;BUMP TABLE POINTER BY 2 (WORD INCREMENT)
3570 021402 077306 SOB R3,4$ ;CONTINUE TO READ & MATCH ALL REGISTERS BEFORE
3571 ;LOADING THE NEXT PATTERN INTO NEXT REGISTER
3572
3573 021404 077424 SOB R4,3$ ;LOOP UNTIL ALL PATTERNS ARE TESTED
3574 021406 000417 BR 24$ ;TEST COMPLETE ***<< NO ERRORS >>***
3575
3576 ;--PREPARE THE FAILURE MESSAGE --
3577
3578 021410 116237 003062 002310 5$: MOVB RESFT3(R2),GDATA ; GET THE EXPECTED RESULT FROM TABLE
3579 021416 117237 002352 002312 MOVB @BSEL(R2),BDATA ; GET THE FAILED BYTE
3580 021424 004737 004434 JSR PC,GETBSR ;GET THE BSEL REGISTERS FOR DUMPING
3581 021430 006202 ASR R2 ;CONVERT WORD OFFSET TO BYTE CSR ADDRESS
3582 021432 010237 002334 MOV R2,REGNUM ;GET THE REGISTER THAT FAILED
3583 021436 GEDF EM6,ERR2 ;ERROR **** DEVICE FATAL ****
; "DEVICE FATAL" ERROR # 11
; TRAP C$ERDF
; .WORD 11
; .WORD EM6
; .WORD ERR2
021436 104455
021440 000013
021442 014540
021444 005304
3584 021446 24$: ENDTST L10027: TRAP C$ETST
021446
021446 104401

```

TEST 4 CSR REGISTERS DATA READ/WRITE

3598

.SBTTL TEST 4 -- CSR REGISTERS DATA READ/WRITE

```

;*****
;*
;* TEST 4 -- CSR REGISTERS DATA READ/WRITE
;*
;* WRITE, READ, AND COMPARE EACH BYTE OF DATA PATTERN B INTO REGISTER BSELO.
;* THEN, REPEAT THIS USING EACH OF THE REMAINING CSR'S, BSEL1-BSEL17. WHEN BSEL1
;* IS BEING TESTED, THE PROGRAM ALWAYS SETS BIT 7 IN THE DATA PATTERN SO THAT
;* RUN WILL NOT BE CLEARED, AND IT ALWAYS CLEARS BIT6 SO THAT MCLR WILL NOT BE
;* SET.
;*
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
;*
;*****

```

```

021450
3599 021450 004737 003762
3600
3601
3602
3603 021454 103002
3604 021456
    021456 104460
3605 021460 000453
3606
3607
3608
3609
3610
3611
3612
3613 021462 005001
3614 021464 005002
3615 021466 016103 002526
3616 021472 062702 000002
3617
3618 021476 113777 000101 160650
3619
3620 021504 116137 002526 002310
3621
3622
3623
3624
3625
3626
3627 021512 022702 000002
3628 021516 001003
3629
3630
3631
3632 021520 142737 000300 002310
3633
3634 021526 113772 002310 002352
3635 021534 123772 002310 002352

```

```

;
; BGNTS1
;
; JSR PC,MSTCLR ;CALL MAINTENANCE READY INITIALIZATION. IF
;               ;MSTCLR SHOULD FAIL BECAUSE THE MRDY FLAG DOES
;               ;NOT BECOME SET, A DEVICE FATAL ERROR WILL BE
;               ;REPORTED, AND MSTCLR WILL SET THE "C" BIT
;               ;IF NO ERROR, PROCEED
;               ;ELSE, REPORT IT AND
;               ;
;               ; TRAP C$ERROR
;               ;
; BR 24$ ; EXIT THIS TEST
;
; NOTE - THE FIRST BYTE LOCATION OF THE PATTERN B TABLE, USED IN THIS TEST,
;        CONTAINS THE NUMBER OF TEST PATTERNS OF PATTERN B TABLE, NOT A
;        TEST PATTERN.
;
; FIRST, INITIALIZE INDEX AND COUNT REGISTERS
8$: CLR R1 ;R1 IS THE 'PATB' INDEX REGISTER
    CLR R2 ;R2 IS THE CSR INDEX REGISTER
    MOV PATB(R1),R3 ;R3 CONTAINS THE NUMBER OF BYTES IN PATB
    ADD #2,R2 ;MOVE POINTER TO FIRST BYTE OF DATA
3618 MOVB 101,08BSEL1 ;STOP THE MICRO-PROCESSOR!!!
3620 MOVB PATB(R1),GDATA ;GET THE PATB DATA BYTE, WE ARE TO USE
; DON'T GET CAUGHT BY THE NEXT INSTRUCTION! "R2" IS AN OFFSET INTO A
; WORD TABLE WHICH CONTAINS THE ADDRESSES OF THE CSR'S. THEREFORE, WHEN
; R2 = 0 - IT POINTS TO BSELO'S ADDRESS, AND WHEN R2 = 2 IT POINTS TO
; BSEL1'S ADDRESS.
3627 CMP #2,R2 ;IS "BSEL1" BEING TESTED?
3628 BNE 2$ ;IF YES, ALTER PATB DATA SO THAT BIT 7 IS
; ALWAYS SET, AND BIT6 IS ALWAYS RESET.
; ELSE, USE PATB DATA AS IS.
3632 BICB #RUN!MCLR,GDATA ;FORCE PATTERN TO RESET BITS 7 & 6
3634 MOVB GDATA,08BSELO(R2) ;PUT PATB DATA INTO REGISTER BEING TESTED
3635 CMPB GDATA,08BSELO(R2) ;COMPARE PATTERN JUST WRITTEN

```

TEST 4 CSR REGISTERS DATA READ/WRITE

```

3636 021542 001414      BEQ      5$          ;TEST PASSES IF A MATCH. ELSE, DEVICE FATAL ERROR
3637
3638
3639                    ; PREPARE FOR THE FAILURE PRINTOUT--
3640 021544 010237 002334      MOV      R2,REGNUM    ;GET THE REGISTER THAT FAILED
3641 021550 117237 002352 002312  MOVB     @BSEL0(R2),BDATA ;SCORE THE BAD DATA
3642 021556 004737 004434      JSR      PC,GETBSR    ;GET THE BSEL REGISTERS FOR DUMPING
3643 021562                    GEDF     EM7,ERR2     ;REPORT ERROR AND EXIT THE TEST
                                ; "DEVICE FATAL" ERROR # 12
                                TRAP     C$ERDF
                                .WORD   12
                                .WORD   EM7
                                .WORD   ERR2
    021562 104455
    021564 000014
    021566 014564
    021570 005304
3644 021572 000406      BR       24$
3645
3646 021574 005201      5$:     INC      R1          ;MOVE TABLE POINTER
3647 021576 077336      SOB     R3,1$         ;DECREMENT NUMBER OF PATTERNS LEFT. IF ZERO,EXIT.
3648                    ;ELSE, CONTINUE TO PATTERN TEST REGISTER
3649 021600 005722      TST     (R2)+         ;INCREMENT THE REGISTER INDEX BY 2
3650 021602 020227 000040      CMP     R2,@<CSREGS*2> ;COMPARE REGISTER INDEX TO NUMBER OF CSR S
3651 021606 101336      BHI     1$           ;IF R2 > 17, END THE TEST
3652
3653 021610      24$:
3654 021610      ENDTST
    021610
    021610 104401
                                L10030:
                                TRAP     C$ETST

```

TEST 5 BASIC MASTER CLEAR

3662

.SBTTL TEST 5 BASIC MASTER CLEAR

: *
: * TEST 5 -- BASIC MASTER CLEAR
: *
: * PERFORM INITIAL MASTER CLEAR. WRITE 356 INTO BSELO AND READ AND CHECK IT.
: * THEN, ISSUE A MASTER CLEAR AND READ AND CHECK BSELO FOR 000.
: *
: * *****

: BGNTST

:
: T5:
: ISSUE A MASTER CLEAR, AND DELAY FOR MICRO DIAGNOSTICS TO COMPLETE BY CALLING
: SUBROUTINE MASCLR.

021612

3663

3664

3665

3666 021612 004737 003614

3667

3668

3669 021616 103002

3670 021620

021620 104460

3671 021622 000441

3672

3673 021624 112777 000356 160520 8%:

3674 021632 122777 000356 160512

3675 021640 001011

3676

3677

3678

3679

3680

3681 021642 004737 003614

3682

3683

3684 021646 103002

3685 021650

021650 104460

3686 021652 000425

3687

3688 021654 122777 000000 160470 9%:

3689

3690 021662 001421

3691

3692 021664 112737 000356 002310 2%:

3693 021672 117737 160454 002312

3694 021700 004737 004434

3695 021704 105077 160442

3696 021710 012737 000000 002334

3697 021716

021716 104455

021720 000015

021722 014515

021724 005304

3698 021726 105077 160420 24%:

3699 021732

```
JSR PC,MASCLR ; ATTEMPT- TO RUN THE MICRO DIAGNOSTIC
; FAILURES WILL BE REPORTED BY THE SUBROUTINE
; AS DEVICE FATAL AND THE "C" BIT WILL BE SET
; IF NO ERROR, PROCEED
; ELSE, REPORT IT AND TRAP C$ERROR

BCC 8$
ERROR

BR 24$ ; EXIT THIS TEST

MOV# 356,BSELO ; # SET BSEL TO ALMOST ALL ONES
CMPB #356,BSELO ; # COMPARE
BNE 2$ ; A MISMATCH INDICATES A DEVICE FATAL ERROR

; ISSUE A MASTER CLEAR, AND DELAY FOR MICRO-DIAGNOSTICS TO COMPLETE BY CALLING
; SUBROUTINE MASCLR.

JSR PC,MASCLR ; ATTEMPT TO RUN THE MICRO-DIAGNOSTIC
; FAILURES WILL BE REPORTED BY THE SUBROUTINE
; AS DEVICE FATAL AND THE "C" BIT WILL BE SET
; IF NO ERROR, PROCEED
; ELSE, REPORT IT AND TRAP C$ERROR

BR 24$ ; EXIT THIS TEST

CMPB #000,BSELO ; THIS REGISTER SHOULD BE ZEROED DURING
; INITIALIZATION
; IF ZERO, *** TEST PASSES ***. ELSE REPORT ERROR
; - PREPARE FOR THE FAILURE PRINTOUT
; # ALMOST ALL ONES IS EXPECTED DATA
; SOMETHING OTHER THAN ALL ONES WAS FOUND. SCORE IT.
; GET THE BSEL REGISTERS FOR DUMPING
; DISABLE INTERRUPTS AS A PRECAUTIONARY MEASURE
; GET THE REGISTER THAT FAILED
; REPORT DEVICE FATAL ERROR
; DEVICE FATAL" ERROR # 13 -
TRAP C$ERRDF

MOV# 356,GDATA ; # ALMOST ALL ONES IS EXPECTED DATA
MOV# BSELO,BDATA ; SOMETHING OTHER THAN ALL ONES WAS FOUND. SCORE IT.
JSR PC,GETBSR ; GET THE BSEL REGISTERS FOR DUMPING
CLRB BSELO ; DISABLE INTERRUPTS AS A PRECAUTIONARY MEASURE
MOV #0,REGNUM ; GET THE REGISTER THAT FAILED
GEDF EM5,ERR2 ; REPORT DEVICE FATAL ERROR
; DEVICE FATAL" ERROR # 13 -
TRAP C$ERRDF
.WORD 13
.WORD EM5
.WORD ERR2

CLRB BSELO ; DISABLE INTERRUPTS AS A PRECAUTIONARY MEASURE
ENDISI
```

J8

CNDMA40 DMV11 MCTRL DIAG #1 MACRO M1200 22-FEB 84 15:22 PAGE 55-1

SEQ 0100

TEST 5 BASIC MASTER CLEAR

021732
021732 104401

L10031: TRAP C\$ETST

TEST 6 BUS RESET

3708

.SBTTL TEST 6 - BUS RESET

```

;*****
;*
;* TEST 6 -- BUS RESET
;*
;* PERFORM AN INITIAL MASTER CLEAR. WRITE 377 INTO BSELO AND READ AND CHECK IT.
;* THEN, ISSUE A RESET INSTRUCTION, STALL FOR COMPLETION, AND READ AND CHECK
;* BSELO FOR 000.
;*
;*****

```

```

;
; BGNTST
;
3709 021734 032737 000001 002350 BIT #BIT0,PFLAG ;IF BUS RESETS ARE NOT ALLOWED, T6::
3710 021742 001072 BNE 24$ ; BYPASS THIS TEST
3711
3712 ; ELSE, ISSUE A MASTER CLEAR, AND DELAY FOR MICRO-DIAGNOSTICS TO COMPLETE BY
3713 ; CALLING SUBROUTINE MASCLR.
3714
3715 021744 004737 003614 JSR PC,MASCLR ; -ATTEMPT- TO RUN THE MICRO DIAGNOSTIC
3716 ; FAILURES WILL BE REPORTED BY THE SUBROUTINE
3717 ; AS DEVICE FATAL AND THE "C" BIT WILL BE SET
3718 021750 103002 BCC 8$ ;IF NO ERROR, PROCEED
3719 021752 ERROR ;ELSE, REPORT IT AND
3720 021754 000465 BR 24$ ; EXIT THIS TEST TRAP C$ERROR
3721
3722 021756 112777 000377 160366 8$: MOVB #377,@BSELO ;SET ALL BITS IN BSELO
3723 021764 122777 000377 160360 CMPB #377,@BSELO ;COMPARE TO ALL BITS SET
3724 021772 001035 BNE 1$ ;A MISMATCH IS A DEVICE FATAL ERROR
3725
3726 021774 BRESET ;FORCE AN EXTERNAL BUS RESET. THIS SHOULD TRAP C$RESET
3727 021774 104433 ;CAUSE BSELO=0 IN ABOUT 100 MICROSECONDS
3728
3729 ; DELAY ABOUT 500 MILLISECONDS FOR THE MICRODIAGNOSTIC TO COMPLETE
3730
3731 021776 013701 002316 MOV DELAY1,R1 ;INITIALIZE COUNTER
3732 022002 105737 002317 2$: TSTB DELAY1+1 ;THIS IS A DUMMY INSTRUCTION TO LENGTHEN THE DELAY
3733 022006 005301 DEC R1 ;TIME TO GET OUT OF THE DELAY?
3734 022010 001374 BNE 2$ ;NO,
3735
3736 022012 122777 000000 160332 CMPB #000,@BSELO ;YES, CHECK FOR REGISTER TO BE ZERO
3737 022020 001443 BEQ 24$ ;A MISMATCH IS A DEVICE FATAL ERROR
3738 ;ELSE, END TEST.
3739
3740 ; --PREPARE FOR THE FAILURE PRINTOUT-
3741
3742 022022 117737 160324 002312 MOVB @BSELO,BDATA ;GET THE ACTUAL DATA
3743 022030 004737 004434 JSR PC,GETBSR ;GET THE BSEL REGISTERS FOR DUMPING
3744 022034 105077 160312 CLRB @BSELO ;DISABLE INTERRUPTS AS A PRECAUTIONARY MEASURE
3745 022040 012737 000000 002310 MOV #000,GDATA ;GET THE EXPECTED DATA
3746 022046 012737 000000 002334 MOV #0,REGNUM ;GET THE REGISTER THAT FAILED
3747 022054 GEDF EM9,ERR2 ;EXTERNAL BUS RESET FAILURE
; "DEVICE FATAL" ERROR # 14 TRAP C$ERDF
022054 104455

```

TEST 6 BUS RESET

```

022056 000016 .WORD 14
022060 014634 .WORD EM9
022062 005304 .WORD ERR2
3748 022064 000421 BR 24$
3749
3750 022066 117737 160260 002312 1$: MOVB @BSELO,BDATA ;GET THE ACTUAL DATA
3751 022074 004737 004434 JSR PC,GETBSR ;GET THE BSEL REGISTERS FOR DUMPING
3752 022100 105077 160246 CLRB @BSELO ;DISABLE INTERRUPTS AS A PRECAUTIONARY MEASURE
3753 022104 112737 000377 002310 MOVB @377,GDATA ;ALL ONES WAS EXPECTED DATA
3754 022112 012737 000000 002334 MOV @0,REGNUM ;GET THE REGISTER THAT FAILED
3755 022120 GEDF EM8,ERR2 ;BSELO COULD NOT BE SET TO ALL ONES
; "DEVICE FATAL" ERROR # 15
022120 104455 TRAP C$ERDF
022122 000017 .WORD 15
022124 014611 .WORD EM8
022126 005304 .WORD ERR2
3756 022130 105077 160216 24$: CLRB @BSELO ;DISABLE INTERRUPTS AS A PRECAUTIONARY MEASURE
3757 022134 ENDTST1
022134 104401 L10032: TRAP C$ETST

```

TEST 7 CSR, MAINTENANCE MICROCODE INTERACTION

3779

.SBTTL TEST 7 -- CSR, MAINTENANCE MICROCODE INTERACTION

```

;*****
;*
;* TEST 7 -- CSR, MAINTENANCE MICROCODE INTERACTION
;*
;* THIS TEST INVOKES THE MAINTENANCE REQUEST MECHANISM THROUGH WHICH THE LSI-11
;* AND 6502 CAN COMMUNICATE. FIRST, A MASTER CLEAR IS DONE WITH ONLY BIT 0
;* (MREQ) SET IN BSEL1. THE PROGRAM THEN CHECKS FOR THE SETTING OF BSEL2 BIT 7
;* (MRDY) BY THE MAINTENANCE MICROCODE WITHIN ABOUT 50 MICRO-SEC., AND IF MRDY
;* DOES NOT GET SET, AN ERROR IS REPORTED.
;*
;* NEXT, THE PROGRAM LOADS SEL4 WITH 000010 AND BSEL6 WITH 125. THEN, ALL CSR'S
;* ARE READ AND CHECKED FOR EXPECTED CONTENTS.
;*
;* BSEL2 IS THEN LOADED WITH A WRITE COMMAND, WHICH SHOULD CAUSE THE MICROCODE
;* TO TRANSFER THE 125 INTO BSEL0. ALL CSR'S ARE THEN READ AND CHECKED FOR
;* EXPECTED CONTENTS.
;*
;* THEN, THE PROGRAM LOADS 252 INTO BSEL0 AND READS AND CHECKS ALL CSR'S. BSEL2
;* IS THEN LOADED WITH A READ COMMAND, WHICH SHOULD CAUSE THE MICROCODE TO
;* TRANSFER THE 252 INTO BSEL6. ALL CSR'S ARE READ AND CHECKED.
;*
;*****

```

```

;
; BGNTST
;
3780 022136 BGNSUB T7::
022136
022136 T7.1:
022136 104402 TRAP C$BSUB
3781 022140 004737 003762 JSR PC,MSTCLR ;PUT THE MICROPROCESSOR IN THE MAINTENANCE LOOP
3782 022144 103003 BCC 10$ ;IF NO ERROR, PROCEED
3783 022146 ERROR ;ELSE, REPORT IT AND
022146 104460 TRAP C$ERROR
3784 022150 000137 022642 JMP ENDT7 ; EXIT THIS TEST
3785
3786 022154 012777 000020 160200 10$: MOV #SLT0,@SEL4 ;PUT ADDRESS OF SELECT REGISTER 0 IN 'ADDRESS' REG
3787 022162 012777 000125 160176 MOV #125,@SEL6 ;PUT THE DATA TO BE WRITTEN IN 'DATA' REGISTER
3788
3789 022170 027727 160156 000400 CMP @SELO,#400 ;ONLY "MREQ" SHOULD BE SET
3790 022176 001411 BEQ 1$ ;IF IT IS, PROCEED WITH TESTING
3791 ;ELSE, SETUP FOR (& REPORT) THE ERROR
3792 022200 017737 160146 002312 MOV @SELO,BDATA ; BAD DATA
3793 022206 012737 000400 002310 MOV #400,GDATA ; GOOD DATA
3794 022214 005037 002334 CLR REGNUM ; REG. NUMBER
3795 022220 000451 BR 4$
3796
3797 022222 027727 160130 000200 1$: CMP @SEL2,#200 ;"MRDY" SET? (ALSO CHECKED BY "MSTCLR")
3798 022230 001412 BEQ 2$ ;YES, PROCEED WITH TESTING
3799 022232 017737 160120 002312 MOV @SEL2,BDATA ; BAD DATA
3800 022240 012737 000200 002310 MOV #200,GDATA ; GOOD DATA
3801 022246 012737 000002 002334 MOV #2,REGNUM ; THE REG. THAT FAILED
3802 022254 000433 BR 4$ ;EXIT TEST
3803
3804 022256 027727 160100 000020 2$: CMP @SEL4,#SLT0 ;COMPARE SELECT REGISTER 4 WITH THE ADDRESS SENT
3805 022264 001412 BEQ 3$ ;A MISMATCH IS A DEVICE FATAL ERROR
3806 022266 017737 160070 002312 MOV @SEL4,BDATA ;GET THE BAD DATA

```


TEST 7 -- CSR, MAINTENANCE MICROCODE INTERACTION

```

3807 022274 012737 000020 002310      MOV    #SLTO,GDATA    ;GET THE GOOD DATA
3808 022302 012737 000004 002334      MOV    #4,REGNUM     ;GET THE REGISTER NUMBER WHICH FAILED
3809 022310 000415                      BR     4$
3810
3811 022312 027727 160050 000125 3$:    CMP    @SEL6,#000125  ;COMPARE SELECT REGISTER 6 WITH THE DATA SENT
3812 022320 001415                      BEQ    60$           ;THIS PART OF THE TEST PASSES IF A MATCH IS FOUND
3813 022322 017737 160040 002312      MOV    @SEL6,BDATA   ;GET THE BAD DATA
3814 022330 012737 000125 002310      MOV    #000125,GDATA ;GET THE GOOD DATA
3815 022336 012737 000006 002334      MOV    #6,REGNUM     ;GET THE REGISTER NUMBER
3816
3817
3818
3819                                     ; -PREPARE FOR THE FAILURE PRINTOUT--
3820 022344                               4$:    GEDF   EM7,ERR5     ;ELSE, AN ERROR HAS BEEN FOUND
                                           ; "DEVICE FATAL" ERROR # 16
                                           TRAP   C$ERDF
                                           .WORD 16
                                           .WORD EM7
                                           .WORD ERR5
022344 104455
022346 000020
022350 014564
022352 005552
3821 022354                               60$:   ENDSUB
022354 104403
022354
3822                                     ;***** > P A R T 2 < *****
3823 022356                               BGNSUB
                                           L10034:
                                           TRAP   C$ESUB
022356 104402
                                           T7.2:
                                           TRAP   C$BSUB
3824
3825 022360 112777 000002 157770      MOVB   #WRILOC,@SEL2 ;SEND THE WRITE LOCATION COMMAND
3826
3827 022366 032777 000200 157762      BIT    #200,@SEL2   ;WE SHOULD HAVE IMEDIATLY LOST "MRDY".
3828 022374 001421                      BEQ    5$           ;GOT WHAT WE EXPECTED, WAIT FOR READY AGAIN
3829 022376 017737 157754 002312      MOV    @SEL2,BDATA  ;SOMETHING WRONG, SETUP FOR AND REPORT ERROR
3830 022404 004737 004434                      JSR    PC,GETBSR    ;GET THE BSEL REGISTERS FOR DUMPING
3831 022410 012737 000002 002310      MOV    #002,GDATA   ;EXPECTED DATA
3832 022416 012737 000002 002334      MOV    #2,REGNUM    ;WE WERE TESTING BSEL2
3833 022424                               GEDF   EM15,ERR2
                                           ; "DEVICE FATAL" ERROR # 17
                                           TRAP   C$ERDF
                                           .WORD 17
                                           .WORD EM15
                                           .WORD ERR2
022424 104455
022426 000021
022430 014733
022432 005304
3834 022434                               ESCAPE T:ST
022434 104410
022436 000206
                                           TRAP   C$ESCAPE
                                           .WORD L10033-.
3835
3836 022440 132777 000200 157710 5$:   BITB   #200,@SEL2   ;WAIT FOR "MRDY" TO GO HIGH AGAIN
3837 022446 001774                      BEQ    5$
3838
3839 022450 004737 004576                      JSR    PC,GETWSR    ;WHEN IT DOES, GET CURRENT REGISTER CONTENTS
3840
3841 022454 023727 002246 000525      CMP    WSR0,#000525 ;COMPARE BYTE SELECT REGISTERS 0 AND 1
3842                                     ;REG 0 = 125, REG 1 = 001
3843 022462 001412                      BEQ    6$           ;THIS PART OF THE TEST PASSES IF A MATCH IS FOUND
3844 022464 012737 000525 002310      MOV    #000525,GDATA ;GET THE GOOD DATA
3845 022472 013737 002246 002312      MOV    WSR0,BDATA   ;GET THE BAD DATA
3846 022500 012737 000000 002334      MOV    #0,REGNUM    ;GET THE REGISTER NUMBER
3847 022506 000451                      BR     9$           ;EXIT TEST

```

TEST 7 CSR, MAINTENANCE MICROCODE INTERACTION

```

3848
3849 022510 023727 002250 000200 6: CMP WSR2,0000200 ;COMPARE BYTE SELECT REGISTERS 2 AND 3
3850 ;REG 2 = 200 -- "MRDY" IS SET & COMMAND IS CLEARED
3851 ;REG 3 = 000 SHOULD BE ZEROES.
3852 022516 001412 BEQ 7: ;THIS PART OF THE TEST PASSES IF A MATCH IS FOUND
3853 022520 012737 000200 002310 MOV 0000200,GDATA ;GET THE GOOD DATA
3854 022526 013737 002250 002312 MOV WSR2,BDATA ;GET THE BAD DATA
3855 022534 012737 000002 002334 MOV 02,REGNUM ;GET THE REGISTER NUMBER
3856 022542 000433 BR 9: ;EXIT TEST
3857
3858
3859 022544 023727 002252 000020 7: CMP WSR4,0SLT0 ;SUBROUTINE ATTEMPTED TO ZERO THIS LOCATION.
3860 ;REG 4 = 020, THE 6502 ADDRESS TO PUT DATA
3861 022552 001412 BEQ 8: ;REG 5 = 000, ZEROED BY MSTCLR
3862 022554 012737 000020 002310 MOV 0SLT0,GDATA ;THIS PART OF THE TEST PASSES IF A MATCH IS FOUND
3863 022562 013737 002252 002312 MOV WSR4,BDATA ;GET THE GOOD DATA
3864 022570 012737 000004 002334 MOV 04,REGNUM ;GET THE BAD DATA
3865 022576 000415 BR 9: ;GET THE REGISTER NUMBER
3866 ;EXIT TEST
3867 022600 023727 002254 000125 8: CMP WSR6,0000125 ;REG 6 = 125, THE WRITE DATA
3868 022606 001415 BEQ ENDT7 ;THIS PART OF THE TEST PASSES IF A MATCH IS FOUND
3869 022610 012737 000125 002310 MOV 0000125,GDATA ;GET THE GOOD DATA
3870 022616 013737 002254 002312 MOV WSR6,BDATA ;GET THE BAD DATA
3871 022624 012737 000006 002334 MOV 06,REGNUM ;GET THE REGISTER NUMBER
3872 ;REG 7 = 000, ZEROED BY MSTCLR.
3873
3874 ;- PREPARE FOR THE FAILURE PRINTOUT-
3875
3876 022632 9: GEDF EM7,ERR5 ;REPORT ERROR.
; 'DEVICE FATAL' ERROR # 18
022632 104455 TRAP C$ERDF
022634 000022 .WORD 18
022636 014564 .WORD EM7
022640 005552 .WORD ERR5
3877 022642 ENDT7: ENDSUB
022642
022642 104403 L10035: TRAP C$ESUB
3878 022644 ENDTST
022644
022644 104401 L10033: TRAP C$ETST

```

3893

.SBTTL TEST 8 RUN FLIP FLOP

```

;*****
;
; TEST 8 -- RUN FLIP-FLOP
;
; THE PROGRAM PUTS THE MICROCODE INTO THE MAINTENANCE LOOP. A 125 CHARACTER
; IS LOADED INTO BSEL6 AND A REQUEST IS MADE TO WRITE THE CONTENTS OF BSEL6
; INTO BSELO. THE PROGRAM THEN READS AND CHECKS BSELO TO CONTAIN 125.
; NEXT, THE RUN FLIP-FLOP IS CLEARED BY LOADING A 0 INTO RUN (BSEL1 BIT 7).
; BSELO IS THEN CLEARED AND THE REQUEST IS MADE AGAIN TO WRITE THE CONTENTS
; OF BSEL6 INTO BSELO. THE PROGRAM STALLS FOR 50 MICRO-SEC. AND CHECKS FOR
; MRDY (BSEL2 BIT 7) NOT SET, AND BSELO STILL CLEARED.
; THEN, THE PROGRAM SETS THE RUN FLIP-FLOP AGAIN BY LOADING A 1 INTO RUN,
; AND CHECKS FOR MRDY SET WITHIN 50 MICRO SEC. AND BSELO = 125.
;
;*****

```

```

;
; BGNTST
;
3894 022646 004737 003762      JSR      PC,MSTCLR      ;CALL SUBROUTINE TO INITIALIZE THE CSR S AND
3895                                ;PUT THE 6502 INTO THE MAINTENANCE LOOP
3896 022652 103003      BCC      2$            ;IF NO ERROR, PROCEED
3897 022654 104460      ERROR                      ;ELSE, REPORT IT AND
3898 022656 000137 023212      JMP      24$            ; EXIT THIS TEST          TRAP      C$ERROR
3899
3900                                ; DO NORMAL WRITE INTO LOCATION USED BY BSELO
3901
3902 022662 004537 004322      2$:      JSR      R5,WRITEI      ;WRITE INTO BSELO THROUGH THE BACK DOOR!
3903 022666 000020                                ; ADDRESS OF BSELO WITHIN RAM
3904 022670 000125      10$:      125                                ; TEST DATA
3905 022672 103002      BCC      5$            ;IF AN ERROR OCCURED.
3906 022674 104460      ERROR                      ;REPORT IT &
3907 022676 000545                                ; EXIT          TRAP      C$ERROR
3908 022700 123777 022670 157444 5$:      CMPB     10$,@BSELO      ;DID THE DATA GO INTO BSELO?
3909 022706 001416      BEQ      11$            ;YES, NOW TRY IT WITH THE "RUN" BIT OFF
3910 022710 017737 157436 002312      MOV      @BSELO,BDATA      ;NO, SETUP & PRINT ERROR MESSAGE
3911 022716 013737 022670 002310      MOV      10$,GDATA
3912 022724 005037 002334      CLR      REGNUM          ;WE'RE SINGLING OUT SELO FOR THE MESSAGE
3913 022730                                ; "DEVICE FATAL" ERROR # 19
3914 022730 104455                                ; IF THIS WRITE DIDN'T WORK, THERE IS NO SENSE
3915 022732 000023                                ; IN TRYING IT WITH "RUN" OFF!
3916 022734 015013                                ;CLEAR BSELO AGAIN
3917 022736 005552                                ;REG'S ARE ALREADY SETUP FROM PREVIOUS WRITE
3918 022740 104410      ESCAPE  TST          ;TURN OFF THE RUN BIT & HOPEFULLY THE 6502 ALSO
3919 022742 000250                                ;TELL MLOOP TO WRITE AGAIN
3919 022750 142777 000200 157376      BICB     @RUN,@BSEL1
3920 022756 112777 000002 157372      MOVB     @WRILOC,@BSEL2
3921

```

TEST 8 - RUN FLIP-FLOP

```

3922 ;AT THIS POINT, THE DMV-11 SHOULD BE QUIESCENT. IT SHOULDN'T EVEN KNOW THAT
3923 ;WE HAVE CLEARED "MRDY". THEREFOR, THERE SHOULD BE NO CHANGE IN ANY OF THE
3924 ;REGISTERS FOR AS LONG AS WE CARE TO WAIT AND WATCH. THE ONLY CHANGE THAT WE
3925 ;COULD POSSIBLY SEE WOULD BE "MRDY" BEING SET AGAIN AND/OR BSELO GETTING
3926 ;LOADED WITH THE TEST DATA -- WHICH IS WHAT THE REQUESTED COMMAND SHOULD
3927 ;ACCOMPLISH WHEN THE FUNCTION IS ALLOWED TO RUN AGAIN.
3928
3929 022764 013701 002320      MOV      DELAY2,R1      ;SETUP AND WAIT FOR A WHILE.
3930 022770 132777 000200 157360 12$: BITB     @MRDY,@BSEL2 ;WHILE WE'RE WAITING, WE MAY AS WELL CHECK 'MRDY'
3931 022776 001042          BNE      14$          ;IF IT GETS SET, WE HAVE AN ERROR BECAUSE
3932 ;NOTHING WAS SUPPOSED TO HAPPEN WITHIN
3933 ;THE 6502 MICRO-PROCESSOR
3934 023000 105777 157346          TSTB     @BSELO      ;WHILE WE'RE AT IT, WE MAY AS WELL LOOK AT
3935 023004 001063          BNE      15$          ;BSELO. THAT ALSO ISN'T SUPPOSED TO CHANGE.
3936 023006 077110          SOB      R1,12$     ;DECREMENT AND CHECK COUNTER -- LOOP TILL DONE
3937
3938 ;IF EVERYTHING GOES OK, WE SHOULD FALL OUT OF THE LOOP TO HERE. OTHERWISE,
3939 ;"MRDY" OR BSELO COULD CHANGE SENDING US TO "14$" OR "15$" RESPECTIVELY TO
3940 ;PRINT AN APPROPRIATE (WE HOPE) ERROR MESSAGE.
3941
3942 ;IF WE DO GET TO HERE, WE CAN NOW SET "RUN" AND THE MLOOP SHOULD PERFORM THE
3943 ;REQUESTED FUNCTION.
3944
3945 023010 152777 000200 157336          BISB     @RUN,@BSEL1 ;SET 'RUN' AND ALLOW THE 6502 TO RUN AGAIN
3946
3947 ;NOW ALL WE HAVE TO DO IS WAIT AGAIN AS BEFORE. EXCEPT THAT THIS TIME "MRDY
3948 ;OR BSELO GETTING SET IS THE VALID CONDITION NOT THE ERROR. FAILURE TO
3949 ;PERFORM IS NOW THE ERROR WE'RE LOOKING FOR.
3950
3951 023016 013701 002320      MOV      DELAY2,R1      ;SETUP AND WAIT FOR A WHILE.
3952 023022 132777 000200 157326 13$: BITB     @MRDY,@SEL2 ;WHILE WE'RE WAITING, "MRDY" SHOULD GO NON ZERO
3953 023030 001070          BNE      24$          ;IF IT GETS SET, WE CAN ASSUME THAT SOMETHING
3954 ;COMPLETED. AT LEASE WE WERE ABLE TO GET THE
3955 ;6502 MICRO-PROCESSOR RUNNING AGAIN
3956 023032 077105          SOB      R1,13$     ;DECREMENT AND CHECK COUNTER - LOOP TILL DONE
3957
3958 ;IF WE GET HERE, WE WEREN'T ABLE TO RESTORE THE 6502 TO A RUNNING STATE!
3959
3960 023034 117737 157316 002312      MOVB     @SEL2,BDATA ;SETUP FOR THE ERROR MESSAGE -- GET BAD DATA
3961 023042 004737 004434      JSR      PC,GETBSR  ;GET THE BSEL REGISTERS FOR DUMPING
3962 023046 117737 002312 002310      MOVB     BDATA,GDATA ;PICK THE REGISTER'S DATA. THE ONLY DIFFERENCE
3963 023054 152737 000200 002310      BISB     @MRDY,GDATA ;BETWEEN GOOD & BAD IS THE "MRDY" BIT
3964 023062 012737 000002 002334      MOV      @2,REGNUM  ;INDICATE THAT WE'RE LOOKING AT BSEL2
3965 023070          GDF      EM17,ERR2 ;NOW REPORT THE ERROR
3966          ;          "DEVICE FATAL" ERROR # 20
3967          TRAP      C$ERDF
3968          .WORD     20
3969          .WORD     EM17
3970          .WORD     ERR2
3971 023100          ESCAPE  TST          ;EXIT TEST (OR LOOP, MAYBE?)
3972          TRAP      C$ESCAPE
3973          .WORD     L10036
3974
3975 ;IF WE GET HERE, BSEL2 CHANGED WHEN THE 6502 WASN'T SUPPOSED TO BE RUNNING!
3976
3977 023104 117737 157246 002312 14$: MOVB     @BSEL2,BDATA ;GET THE UNEXPECTEDLY ALTERED CONTENTS OF BSEL2
3978 023112 004737 004434      JSR      PC,GETBSR  ;GET THE BSEL REGISTERS FOR DUMPING

```


TEST 9 LOW RAM (00 OF) SCRATCHPAD

4004

.SBTTL TEST 9 - LOW RAM (00 OF) SCRATCHPAD

```

;*****
;*
;* TEST 9 -- LOW RAM (00-OF) SCRATCHPAD
;*
;* THIS TEST FIRST PERFORMS AN ADDRESSING TEST OF RAM LOCATIONS (00 OF), BY
;* WRITING THE ADRS INTO EACH LOCATION AND AFTER EACH WRITE, ALL THE LOCATIONS
;* ARE READ AND CHECKED FOR EXPECTED CONTENTS.
;*
;* THEN, THE TEST PERFORMS READ/WRITE DATA TESTING OF RAM LOCATIONS 00-OF,
;* BY WRITING, READING, AND COMPARING ALL BYTES OF DATA PATTERN B IN EACH
;* LOCATION.
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
;*
;*****

```

; BGNTST

```

;*****
;
; T9::
4005 023214 004737 003762 JSR PC,MSTCLR ;INIT DMV & ENTER M LOOP
4006 023220 103003 BCC 1$ ;IF NO ERROR, PROCEED WITH TESTING
4007 023222 ERROR ;ELSE, REPORT ERROR
; TRAP C$ERROR
4008 023224 ESCAPE TST ; & EXIT TEST
; TRAP C$ESCAPE
; .WORD L10037
4009 023230 012737 000001 002444 1$: MOV #1,TMP2 ;DATA GENERATION ALGORITHM CODE
4010 023236 012737 003777 002474 MOV #2047.,TMPE ;LAST VALID ADDRESS
4011 023244 004737 023402 2$: JSR PC,T9.RST ;RESET TMP3, TMPA, & TMPF
4012 023250 005037 002450 CLR TMP4 ;TEST DATA
4013 023254 005037 002452 CLR TMP5 ;ACTUAL DATA
4014
4015 ; IN THIS PHASE OF TESTING WE WRITE, READ & CHECK EACH LOCATION INDIVIDUALLY.
4016
4017 023260 004737 023420 4$: JSR PC,WRCRAM ;WRITE, READ, & CHECK 1 BYTE OF RAM
4018 023264 103003 BCC 5$ ;IF NO ERROR, PROCEED
4019 023266 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
4020 023270 ESCAPE TST ; & LOOP IF ERROR
; TRAP C$ESCAPE
; .WORD L10037-
4021 023274 005237 002464 5$: INC TMPA ;POINT TO NEXT LOCATION
4022 023300 023737 002464 002474 CMP TMPA,TMPE ;HAVE WE TESTED ALL OF RAM?
4023 023306 101764 BLOS 4$ ;NO, TEST ANOTHER BYTE
4024 023310 BREAK ;ELSE, SEE IF A +C HAS BEEN STRUCK
; TRAP C$BRK
4025 ;THEN PROCEED TO THE NEXT PHASE OF TESTING
4026
4027 ; IN THIS PHASE OF TESTING WE READ & CHECK DATA WHICH SHOULD ALREADY BE IN
4028 ; EACH LOCATION OF RAM BEING CHECKED.
4029
4030 023312 004737 023402 8$: JSR PC,T9.RST ;RESET TMP3, TMPA, & TMPF
4031 023316 004737 023616 JSR PC,RCRAM ;READ & CHECK 1 BYTE OF RAM
4032 023322 103001 BCC 9$ ;IF NO ERROR, PROCEED
4033 023324 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
;*****

```

TEST 9 -- LOW RAM (00-0F) SCRATCHPAD

```

4034 023326 005237 002464 9$: INC TMPA ;POINT TO NEXT LOCATION
4035 023332 023737 002464 002474 CMP TMPA, TMPE ;HAVE WE TESTED ALL OF RAM?
4036 023340 101766 BLOS 8$ ;NO, TEST ANOTHER BYTE
4037 023342 BREAK ;ELSE, SEE IF A 'C HAS BEEN STRUCK
023342 104422 ;THEN PROCEED TO THE NEXT PHASE OF TESTING TRAP C$BRK
4038 ;
4039 ;
4040 023344 005037 007104 CLR ER47CT ;RESET ERROR PRINT COUNT
4041 023350 005237 002444 INC TMP ; ADVANCE TO NEXT DATA GEN. ALGORITHM CODE
4042 023354 023727 002444 000007 CMP TMP2, #7 ;HAVE WE DONE ALL THE CODES WE'RE GOING TO DO?
4043 023362 002730 BLT 2$ ;NO, THEN GO DO THIS PATTERN IN RAM
4044 023364 004537 004322 JSR R5,WRITEI ;ELSE, CLEAR RAM LOCATION 00B3 (HEX) & EXIT
4045 023370 000173 ; (THIS CONVERTS TO 00B3 HEX.)
4046 023372 000000 0 ; (THIS WE HOPE, WILL CLEAR IT)
4047 023374 103001 BCC .+4 ;IF NO ERROR, PROCEED
4048 023376 ERROR ;ELSE, REPORT IT
023376 104460 TRAP C$ERROR
4049 023400 ENDTST L10037:
023400 TRAP C$ETST
023400 104401 ;
4050 ; RESET THE FOLLOWING THREE REGISTERS
4051 ;
4052 ;
4053 023402 005037 002446 T9.RST: CLR TMP3 ;TEST DATA PATTERN INDEX
4054 023406 005037 002464 CLR TMPA ;RAM LOCATION ADDRESS
4055 023412 005037 002476 CLR TMPF ;RESET ALL ERROR FLAGS
4056 023416 000207 RTS PC
4057 ;
4058 ; WRITE, READ, & CHECK ONE LOCATION
4059 ;
4060 023420 010046 WRCRAM: MOV RO, -(SP) ;SAVE WORKING REGISTERS
4061 ;
4062 023422 004737 023776 JSR PC, PATGEN ;GENERATE ONE DATA PATTERN BYTE
4063 ;
4064 023426 013700 002464 MOV TMPA, RO ;GET ADDRESS WHERE WE CAN CHECK IT MORE EASILY
4065 023432 020027 000020 CMP RO, #SLTO ;IS ADDRESS BELOW THE SELECT REGISTER AREA?
4066 023436 103412 BLO 2$ ;YES, GOOD. IT CAN BE TESTED.
4067 023440 020027 000030 CMP RO, #SLTO+8. ;IS IT ABOVE THE SELECT REGISTER AREA?
4068 023444 103007 BHIS 2$ ;YES, GOOD. IT CAN BE TESTED.
4069 023446 023727 002444 000006 CMP TMP2, #6 ;NO, IF "INCREMENTAL", BACK UP INDEX
4070 023454 001055 BNE 12$ ;ELSE JUST BYPASS TEST
4071 023456 005337 002446 DEC TMP3 ;DECREMENT INDEX TO WHAT IT WAS BEFORE "PATGEN"
4072 023462 000452 BR 12$ ; AND THEN BYPASS THE TESTING
4073 ;
4074 023464 010037 023500 2$: MOV RO, 4$ ;SETUP ALL POINTERS FOR THE CURRENT RAM LOCATION
4075 023470 010037 023512 MOV RO, 8$
4076 ;
4077 023474 004537 004310 JSR R5,WRITE ;WRITE ONE BYTE OF THE TEST DATA
4078 023500 000000 4$: .WORD 0 ;**** MODIFIED FROM ABOVE ****
4079 023502 002450 TMP4 ;TEST DATA IS IN TMP4
4080 023504 103442 BCS 14$ ;IF ERROR WRITING, FORGET THE REST
4081 ;
4082 023506 004537 004064 JSR R5,READ ;READ THAT BYTE BACK AGAIN
4083 023512 000000 8$: .WORD 0 ;**** MODIFIED FROM ABOVE ****
4084 023514 002452 TMP5
4085 023516 103435 BCS 14$ ;IF ERROR READING, FORGET THE REST
4086 ;

```

TEST 9 LOW RAM (00 OF) SCRATCHPAD

4087	023520	123737	002450	002452		CMPB	TMP4,TMP5		;DID WE READ WHAT WE WROTE?
4088	023526	001430				BEQ	12\$;YES, EXIT
4089	023530	132737	000002	002476		BITB	#BIT1,TMPF		;NO, HAVE WE ALREADY DONE THIS ERROR'S HEADER?
4090	023536	001020				BNE	9\$;YES, ONLY REPORT DATA
4091	023540	112737	000002	002476		MOVB	#BIT1,TMPF		;ELSE, CALL MONITOR & PRINT HEADING
4092	023546					GTFD	EM47A,ERR47		;QUEUE UP THE ERROR MESSAGE
	023546	012737	000001	002236					; QUEUE "DEVICE FATAL" ERROR # 23
	023554	012737	000027	002240				MOV	#T.EDF,ERRTYP
	023562	012737	015675	002242				MOV	#23,ERRNBR
	023570	012737	006724	002244				MOV	#EM47A,ERRMSG
	023576	000402						MOV	#ERR47,ERRBLK
4093	023576	000402				BR	10\$		
4094									
4095	023600	004737	007110		9\$:	JSR	PC,ERR47.		;JUST PRINT DATA
4096	023604	000261			10\$:	SEC			; & SET THE ERROR FLAG
4097	023606	000401				BR	14\$; & GO DIRECTLY TO THE EXIT "RTS"
4098									
4099	023610	000241			12\$:	CLC			;NORMAL EXIT - MAKE SURE THE ERROR FLAG IS CLEAR
4100	023612	012600			14\$:	MOV	(SP)+,RO		;RESTORE WORK REGISTERS
4101	023614	000207				RTS	PC		
4102									
4103	023616	010046			RCRAM:	MOV	RO,-(SP)		;SAVE WORKING REGISTERS
4104									
4105	023620	004737	023776			JSR	PC,PATGEN		;GENERATE ONE DATA PATTERN BYTE
4106									
4107	023624	013700	002464			MOV	TMPA,RO		;GET ADDRESS WHERE WE CAN CHECK IT MORE EASILY
4108	023630	020027	000020			CMP	RO,#SLTO		;IS ADDRESS BELOW THE SELECT REGISTER AREA?
4109	023634	103412				BLO	2\$;YES, GOOD. IT CAN BE TESTED.
4110	023636	020027	000030			CMP	RO,#SLTO+8.		;IS IT ABOVE THE SELECT REGISTER AREA?
4111	023642	103007				BHIS	2\$;YES, GOOD. IT CAN BE TESTED.
4112	023644	023727	002444	000006		CMP	TMP2,#6		;NO, IF "INCREMENTAL", BACK UP INDEX
4113	023652	001046				BNE	12\$;ELSE JUST BYPASS TEST
4114	023654	005337	002446			DEC	TMP3		;DECREMENT INDEX TO WHAT IT WAS BEFORE "PATGEN"
4115	023660	000443				BR	12\$; AND THEN BYPASS THE TESTING
4116									
4117	023662	010037	023672		2\$:	MOV	RO,8\$;SETUP POINTER FOR THE CURRENT RAM LOCATION
4118									
4119	023666	004537	004064			JSR	R5,READ		;READ THAT BYTE BACK AGAIN
4120	023672	000000			8\$:	.WORD	0		;**** MODIFIED FROM ABOVE ****
4121	023674	002452				TMP5			
4122	023676	103435				BCS	14\$;IF ERROR READING, FORGET THE REST
4123									
4124	023700	123737	002450	002452		CMPB	TMP4,TMP5		;WAS THIS LOC. STILL OK?
4125	023706	001430				BEQ	12\$;YES, EXIT
4126	023710	132737	000004	002476		BITB	#BIT2,TMPF		;NO, HAVE WE ALREADY DONE THIS ERROR S HEADER?
4127	023716	001020				BNE	9\$;YES, ONLY REPORT DATA
4128	023720	112737	000004	002476		MOVB	#BIT2,TMPF		;ELSE, CALL MONITOR & PRINT HEADING
4129	023726					GTFD	EM47B,ERR47		;QUEUE UP THE ERROR MESSAGE
	023726	012737	000001	002236					; QUEUE "DEVICE FATAL" ERROR # 24
	023734	012737	000030	002240				MOV	#T.EDF,ERRTYP
	023742	012737	015735	002242				MOV	#24,ERRNBR
	023750	012737	006724	002244				MOV	#EM47B,ERRMSG
	023756	000402						MOV	#ERR47,ERRBLK
4130	023756	000402				BR	10\$		
4131									
4132	023760	004737	007110		9\$:	JSR	PC,ERR47.		;JUST PRINT DATA
4133	023764	000261			10\$:	SEC			; & SET THE ERROR FLAG

TEST 9 - LOW RAM (00-0F) SCRATCHPAD

```

4134 023766 000401          BR      14$          ;      & GO DIRECTLY TO THE EXIT "RTS"
4135
4136 023770 000241          12$:   CLC              ;NORMAL EXIT - MAKE SURE THE ERROR FLAG IS CLEAR
4137 023772 012600          14$:   MOV      (SP)+,R0 ;RESTORE WORK REGISTERS
4138 023774 000207          RTS      PC
4139
4140
4141 ;*****
4142 ; PATGEN -- SUBROUTINE TO GENERATE A TEST DATA BYTE FOR A SPECIFIC ELEMENT
4143 ;
4144 ;      CALLING SEQUENCE:
4145 ;
4146 ;      <SET TEST PATTERN CODE # IN "TMP2">
4147 ;      <SET TEST PATTERN INDEX IN "TMP3">
4148 ;      JSR      PC,PATGEN
4149 ;      <NEXT SEQUENTIAL INSTRUCTION>
4150 ;
4151 ;      TEST PATTERN CODES:
4152 ;
4153 ;      1 -- ALL ONES
4154 ;      2 -- ALL ZEROES
4155 ;      3 - 1 BIT ALTERNATING
4156 ;      4 -- 2 BITS ALTERNATING
4157 ;      5 -- ADDRESS IN ADDRESS
4158 ;      6 -- INCREMENTAL (INDEX IN ADDRESS)
4159 ;
4160 ;      THE TEST PATTERN INDEX INDICATES HOW FAR INTO THE TEST PATTERN STRING OF
4161 ;      BYTES WE ARE. I.E. IT SPECIFIES THE NUMBER OF THE BYTE OF THE WHOLE STRING
4162 ;      OF BYTES COMPOSING THE COMPLETE TEST PATTERN.
4163 ;
4164 ;*****
4165
4166 023776          PATGEN:
4167 023776 023727 002444 000000  CMP      TMP2,#2          ;DECODE THE TEST PATTERN IDENTIFIER
4168 024004 002414          BLT      1$              ;0, 1, OR NEGATIVE WILL GIVE 'ALL ONES'
4169 024006 001417          BEQ      2$              ;2 = "ALL ZEROES"
4170 024010 023727 002444 000004  CMP      TMP2,#4
4171 024016 002416          BLT      3$              ;3 = "1 BIT ALTERNATING"
4172 024020 001431          BEQ      4$              ;4 = "2 BIT ALTERNATING PATTERN"
4173 024022 023727 002446 000006  CMP      TMP3,#6
4174 024030 002441          BLT      5$              ;5 = "ADDRESS IN ADDRESS"
4175 024032 001444          BEQ      6$              ;6 = "INCREMENTAL" (INDEX IN ADDRESS)
4176 024034 000404          BR      2$              ;UNDEFINED = "ALL ZEROES"
4177
4178 024036 112737 000377 002450 1$:   MOVB     #377,TMP4        ;"ALL ONES" DATA PATTERN
4179 024044 000443          BR      60$
4180
4181 024046 105037 002450          2$:   CLRB     TMP4          ;"ALL ZEROES" DATA PATTERN
4182 024052 000440          BR      60$
4183
4184 024054 132737 000001 002446 3$:   BITB     #1,TMP3          ;"1 BIT ALTERNATING" PATTERN
4185 024062 001404          BEQ      20$             ;IF EVEN, USE "252"
4186 024064 112737 000125 002450  MOVB     #125,TMP4        ;IF ODD, USE "125"
4187 024072 000430          BR      60$             ; PATTERN: 10101010
4188 024074 112737 000252 002450 20$:  MOVB     #252,TMP4
4189 024102 000424          BR      60$             ; PATTERN: 01010101
4190

```

TEST 9 LOW RAM (00-OF) SCRATCHPAD

```

4191 024104 132737 000001 002446 4$:    BITB    #1,TMP3            ;"2 BIT ALTERNATING" PATTERN
4192 024112 001404                        BEQ    22$            ;IF EVEN, USE "214"
4193 024114 112737 000063 002450                        MOVB   #063,TMP4       ;IF ODD, USE "063"
4194 024122 000414                        BR     60$           ; PATTERN: 11001100
4195 024124 112737 000214 002450 22$:    MOVB   #214,TMP4       ; PATTERN: 00110011
4196 024132 000410                        BR     60$           ; PATTERN: 00110011
4197
4198 024134 113737 002464 002450 5$:    MOVB   TMPA,TMP4       ;"ADDRESS IN ADDRESS"
4199 024142 000404                        BR     60$
4200
4201 024144 113737 002446 002450 6$:    MOVB   TMP3,TMP4       ;"INCREMENTAL" (INDEX IN ADDRESS)
4202 024152 000400                        BR     60$
4203
4204 024154 005237 002446                        INC    TMP3           ;INCREMENT PATTERN INDEX FOR NEXT CALL
4205 024160 000207                        RTS    PC
4206
4207

```

TEST 10 DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)

4279

.SBTTL TEST 10 -- DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)

TEST 10 -- DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)

GENERAL DESCRIPTION:

FIRST, THE 2K BYTE LOCATIONS IN RAM ARE LOADED WITH 0'S (SEE NOTE BELOW). THEN, THE FIRST LOCATION IS READ AND CHECKED, A SINGLE 1 IS WRITTEN INTO THE LOW BIT POSITION, AND THIS IS READ AND CHECKED. THIS IS DONE FOR ALL BYTES IN THE RAM, BY INCREMENTING THE ADDRESS TO POINT TO THE NEXT RAM LOCATION.

THEN, THE NEXT BIT POSITION IS CHOSEN TO INSERT A 1, AND ALL LOCATIONS ARE READ, WRITTEN, AND READ AS BEFORE. THIS IS CONTINUED FOR ALL BIT POSITIONS UNTIL THE ENTIRE RAM IS WRITTEN TO ALL 1'S. THE ABOVE OPERATIONS ARE PERFORMED A SECOND TIME, WITH 0'S INSERTED INTO THE RAM INSTEAD OF 1'S. THIS RESULTS IN THE ENTIRE RAM BEING WRITTEN TO ALL 0'S.

THIS TEST CONSTITUTES A THOROUGH TEST OF THE RAM. IT IS CAPABLE OF DETECTING THE FOLLOWING FAULTS : STUCK ADDRESS BITS, UNI- AND BI-DIRECTIONAL COUPLING BETWEEN ADDRESS BITS, STUCK MEMORY BITS, AND UNI- AND BI-DIRECTIONAL COUPLING BETWEEN MEMORY BITS IN BOTH ROWS AND COLUMNS OF THE MEMORY MATRIX.

NOTE:

THIS TEST DOES NOT CHECK LOCATIONS 0010-001F, SO THAT THE PRIMARY CSR'S ARE NOT WRITTEN. IT DOES TEST LOCATIONS 0000-000F (SCRATCHPAD RAM) AND LOCATIONS 0020-002F (SECONDARY CSR'S), AS WELL AS 0030-0800 (BASIC RAM).

THE "TMP#" REGISTERS ARE USED HERE TO CONTAIN THE VARIOUS CONSTANTS & VARIABLES USED THROUGHOUT THIS TEST. A LIST OF THEIR ASSIGNMENTS SEEMS USEFUL SO HERE IT IS:

- TMP0 POINTS TO THE FIRST LOCATION AFTER THE SELECT REGISTERS.
- TMP1 ----
- TMP2 TEST PATTERN ID CODE UNUSED BY THIS TEST.
- TMP3 TEST DATA PATTERN INDEX -- UNUSED BY THIS TEST.
- TMP4 TEST DATA PATTERN. THE HIGH BYTE IS THE PATTERN BEING WRITTEN ON ANY GIVEN PASS AND THE LOW BYTE IS THE PATTERN THAT WAS WRITTEN BY THE PREVIOUS PASS THROUGH THE RAM.
- TMP5 DATA READ FROM THE RAM. ONLY THE LOW BYTE IS USED.
- TMP6 -
- TMP7 - - -
- TMP8 ----
- TMP9 ----
- TMPA RAM ADDRESS BEING TESTED.
- TMPB BIT POINTER. NUMBER OF THE BIT WITHIN THE DATA FIELD WHICH IS BEING SWITCHED ON EACH WRITE WITHIN THE CURRENT PASS.
- TMPC DATA FLAG. BIT 0 OF THIS WORD IS THE VALUE TO WHICH THE BIT

TEST 10 -- DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)

```

4311          ; ZERO OUT THE REST OF RAM -- ALL LOC'S ABOVE THE SELECT REGISTERS
4312
4313 024266 013737 002440 024312      MOV    TMP0,6$      ;FIRST LOCATION OF TEST AREA (18 HEX)
4314 024274 013703 002474              MOV    TMPE,R3      ;START WITH "LAST ADDR. TO BE TESTED" AND CALC.
4315 024300 163703 002440              SUB    TMP0,R3      ;THE # OF LOCATIONS TO BE TESTED (800 18 (HEX))
4316 024304 005203                      INC    R3           ; (THIS MAKES SURE WE GET EVERY SINGLE BYTE)
4317
4318 024306 004537 004322      4$:   JSR    R5,WRITEI   ;ZERO OUT THE ALL OF THE TEST AREA
4319 024312 000000      6$:   .WORD  0
4320 024314 000000              0
4321 024316 103003              BCC    .+10        ;IF NO ERROR, PROCEED
4322 024320                      ERROR              ;ELSE, REPORT IT
4323 024322                      ESCAPE TST          ; AND EXIT THIS TEST          TRAP    C$ERROR
4324 024322 104410                      .WORD  L10040-.
4325 024324 000614
4326 024326 005237 024312      INC    6$          ;POINT TO NEXT LOCATION
4327 024332 077313      SOB    R3,4$      ;IF MORE TO BE DONE, DO IT
4328 024334 105037 002450      CLRB  TMP4        ;THIS IS WHAT WE JUST SET ALL RAM LOCATIONS TO
4329
4330          ;----- BEGINNING OF OUTER LOOP -----
4331 024340 005037 002472      8$:   CLR    TMPD      ;"SET FWD SEQUENCE" (DIRECTION FLAG)
4332 024344 005037 002466      CLR    TMPB      ;"SET BIT POSITION = 0" (BIT POINTER)
4333          ;"SET ADDRESS = 0" BUT OUR MEMORY STARTS @
4334 024350 005037 002464      CLR    TMPA      ; 18 HEX. SO:
4335 024354 112737 000001 002451      MOVB  #BIT0,TMP4+1 ;INITIALIZE ADDRESS POINTER
4336          ;INITIALIZE CURRENT & NEXT DATA BYTES
4337          ;----- "READ CURRENT ADDRESS" -----
4338
4339 024362 000240      10$:  NOP
4340 024364 000240      NOP
4341 024366          BREAK          ;FIRST SEE IF A +C HAS BEEN STRUCK BY OPERATOR
4342 024370 013737 002464 024402      MOV    TMPA,40$   TRAP    C$BRK
4343 024376 004537 004064      JSR    R5,READ    ;NO, PUT ADDRESS INTO READ CALL
4344 024402 000000      40$:  0              ;GO READ ONE LOCATION
4345 024404 002452          TMP5          ;**** MODIFIED ABOVE **** (ADDRESS)
4346 024406 103003          BCC    .+10        ;ADDRESS OF DATA READ
4347 024410          ERROR          ;IF NO ERROR, PROCEED
4348 024412 104460          ESCAPE TST        ;ELSE, REPORT IT
4349 024412 104410          ; AND EXIT THIS TEST          TRAP    C$ERROR
4350 024414 000524          .WORD  L10040-.
4351          ;----- CHECK DATA (FIRST TIME) -----
4352 024416 000240          NOP
4353 024420 000240          NOP
4354 024422 123737 002452 002450      CMPB  TMP5,TMP4   ;CHECK AGAINST EXPECTED DATA
4355 024430 001421          BEQ    12$        ;IF OK, PROCEED
4356 024432 032737 000006 002476      BIT   #BIT1+BIT2,TMPF ;NO, HAS AN ERROR ALREADY BEEN REPORTED?
4357 024440 001010          BNE   42$        ;YES, JUST PRINT DATA IF WANTED
4358 024442 012737 000002 002476      MOV   #BIT1,TMPF  ;NO, SET FLAG FOR NEXT TIME
4359 024450          GEDF  EM48A,ERR48 ; AND PRINT COMPLETE ERROR MESSAGE
          ; "DEVICE FATAL" ERROR # 25

```

TEST 10 -- DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)

SEQ 0117

```

024450 104455
024452 000031
024454 016016
024456 007632
4360 024460 000405 BR 12$ ;PROCEED WITH TESTING
4361 024462 012737 000002 002476 42$: MOV #BIT1,TMPF ;INDICATE A "PRE" WRITE ERROR
4362 024470 004737 010030 JSR PC,ERR48. ;USE ERROR HANDLER ONLY -- NO HEADER
4363
4364 ;----- WRITE NEW DATA -----
4365
4366 024474 013737 002464 024506 12$: MOV TMPA,44$ ;GET THIS ADDRESS FOR THIS WRITE CALL
4367 024502 004537 004310 JSR R5,WRITE ;WRITE THE UPDATED DATA IN THIS LOCATION
4368 024506 000000 44$: .WORD 0
4369 024510 002451 TMP4+1 ;NEW DATA ELEMENT RESIDES IN TMPD+1
4370 024512 103003 BCC .+10 ;IF NO ERROR, PROCEED
4371 024514 104460 ERROR ;ELSE, REPORT IT
4372 024516 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
024516 104410 TRAP C$ESCAPE
024520 000420 .WORD L10040 .
4373
4374 ;----- RE-"READ CURRENT ADDRESS" -----
4375
4376 024522 013737 002464 024534 MOV TMPA,46$ ;GET ADDRESS FOR THIS READ
4377 024530 004537 004064 JSR R5,READ ;READ DATA JUST WRITTEN
4378 024534 000000 46$: .WORD 0
4379 024536 002452 TMP5
4380 024540 103003 BCC .+10 ;IF NO ERROR, PROCEED
4381 024542 104460 ERROR ;ELSE, REPORT IT
4382 024544 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
024544 104410 TRAP C$ESCAPE
024546 000372 .WORD L10040 .
4383
4384 ;----- CHECK NEW DATA VALUE -----
4385
4386 024550 000240 NOP
4387 024552 000240 NOP
4388 024554 123737 002451 002452 CMPB TMP4+1,TMP5 ;DID THE WRITE WORK CORRECTLY?
4389 024562 001421 #EQ 14$ ;YES, PROCEED WITH TESTING
4390 024564 032737 000006 002476 BIT #BIT1+BIT2,TMPF ;NO, HAS AN ERROR ALREADY BEEN REPORTED?
4391 024572 001010 BNE 48$ ;YES, ONLY USE ERROR HANDLER -- NO HEADER PLEASE
4392 024574 012737 000004 002476 MOV #BIT2,TMPF ;NO, INDICATE THAT WE'RE PRINTING A HEADER HERE
4393 024602 004737 010030 GEDF EM48A,ERR48 ;REPORT RE-WRITE ERROR
; "DEVICE FATAL" ERROR # 26
024602 104455 TRAP C$ERDF
024604 000032 .WORD 26
024606 016016 .WORD EM48A
024610 007632 .WORD ERR48
4394 024612 000405 BR 14$ ;PROCEED WITH TESTING
4395
4396 024614 012737 000004 002476 48$: MOV #BIT2,TMPF ;INDICATE A "POST" WRITE ERROR
4397 024622 004737 010030 JSR PC,ERR48. ;JUST REPORT DATA -- NO HEADER
4398
4399 ;----- "FORWARD SEQUENCE ?" -----
4400
4401 024626 000240 14$: NOP

```

TEST 10 DATA RAM MOVING INVERSIONS (LOC'S 0018-01FF HEX)

```

4402 024630 005737 002472          TST      TMPD          ;CHECK DIRECTION      0 = FORWARD
4403 024634 001056          BNE      26$          ;REVERSE ----> PROCESS REVERSE ADDRESSING
4404                                     ;FORWARD
4405
4406          ; . . . . . PROCESS FORWARD SEQUENCE -- "LAST ADDRESS" . . . . .
4407
4408 024636 000240          16$:  NOP
4409 024640 023737 002464 002474      CMP      TMPA,TMPE    ;WAS THIS ADDR. THE LAST ONE?
4410 024646 001413          BEQ      18$          ;YES, THEN CHECK THE BIT POSITION
4411 024650 005237 002464          50$:  INC      TMPA    ;NO, THEN INCREMENT THE ADDR.
4412
4413          ; HERE WE MAKE SURE THE ADDRESS IS NOT WITHIN THE SELECT REGISTER AREA. IF IT
4414          ; IS, WE WON'T USE IT -- BUT GO BACK AND DECREMENT TO THE NEXT ADDRESS AGAIN.
4415
4416 024654 022737 000020 002464      51$:  CMP      @SLTO,TMPA ;IS IT BELOW THE AREA WE CAN'T CHECK?
4417 024662 101237          BHI      10$          ;YES, THEN WE CAN CHECK THIS LOCATION DO IT
4418 024664 023737 002440 002464      CMP      TMP0,TMPA    ;IS IT BELOW THE BOTTOM ADDRESS?
4419 024672 101633          BLOS    10$          ;NO, TEST THIS LOCATION
4420 024674 000765          BR       50$          ;YES, PERFORM THE INCREMENT AGAIN
4421
4422          ;----- "FWD" SEQUENCE -- "LAST BIT POSITION?" -----
4423
4424 024676 000240          18$:  NOP
4425 024700 005037 010024          CLR      ER48CT      ;RESET ERROR PRINT COUNT
4426 024704 023727 002466 000007      CMP      TMPB,#7     ;DID WE JUST PROCESS THE LAST BIT POSITION?
4427 024712 002016          BGE     20$          ;YES, THEN WERE WE DOING 1'S OR 0'S
4428 024714 005237 002466          INC     TMPB         ;NO, THEN INCREMENT THE BIT COUNTER
4429 024720 005037 002464          24$:  CLR      TMPA     ;RE-INITIALIZE ADDRESS POINTER
4430 024724 113737 002451 002450      57$:  MOV     TMP4-1,TMP4 ;USE "NEXT" DATA AS "CURRENT" DATA
4431 024732 013700 002470          MOV     TMPC,RO      ;USE ONE BIT OF THE "DATA" SWITCH TO
4432 024736 006000          ROR     RO
4433 024740 106137 002451          ROLB   TMP4-1       ;BUILD A NEW "NEXT" DATA VALUE
4434 024744 000137 024362          55$:  JMP     10$          ; & TEST IT
4435
4436          ;----- "FWD" SEQUENCE -- "DATA = 1?" -----
4437
4438 024750 000240          20$:  NOP
4439 024752 005037 002466          CLR     TMPB         ;POINT TO BIT 0,
4440 024756 005137 002470          COM     TMPC         ;SWITCH DATA. IF 1'S, DO 0'S; IF 0'S DO 1'S
4441 024762 001756          BEQ     24$          ;IF WENT TO FORWARD, . . . . .
4442 024764 005137 002472          COM     TMPD         ;SWITCH DIRECTION
4443 024770 000755          BR      57$         ;ELSE, BACKWARD. . . . .
4444
4445          ; . . . . . "BKWD" SEQUENCE "ADDRESS = 0?" . . . . .
4446
4447 024772 000240          26$:  NOP
4448 024774 005737 002464          TST     TMPA         ;HAVE WE JUST PROCESSED THE FIRST ADDRESS?
4449 025000 001413          BEQ     28$          ;YES, CHECK BIT POSITION
4450 025002 005337 002464          52$:  DEC     TMPA     ;NO, DECREMENT THE ADDRESS
4451
4452          ; HERE WE MAKE SURE THE ADDRESS IS NOT WITHIN THE SELECT REGISTER AREA. IF IT
4453          ; IS, WE WON'T USE IT -- BUT GO BACK AND DECREMENT TO THE NEXT ADDRESS AGAIN.
4454
4455 025006 022737 000020 002464      56$:  CMP      @SLTO,TMPA ;IS IT BELOW THE AREA WE CAN'T CHECK?
4456 025014 101031          BHI     58$          ;YES, THEN WE CAN CHECK THIS LOCATION DO IT
4457 025016 023737 002440 002464      CMP     TMP0,TMPA    ;IS IT BELOW THE BOTTOM ADDRESS?
4458 025024 101425          BLOS   58$          ;NO, TEST THIS LOCATION

```

TEST 10 DATA RAM MOVING INVERSIONS (LOC'S 0018 01FF HEX)

```

4459 025026 000765          BR      52$          ;YES, PERFORM THE DECREMENT AGAIN
4460
4461          ;          "BKWD" SEQUENCE - "LAST BIT POSITION"
4462
4463 025030 000240          28$:  NOP
4464 025032 005037 010024    CLR      ER48CT      ;RESET ERROR PRINT COUNT
4465 025036 022737 000007 002466    CMP      #7, TMPB    ;LAST BIT POSITION?
4466 025044 003417          BLE      30$          ;YES, CHECK DATA
4467 025046 005237 002466    INC      TMPB        ;NO, INCREMENT BIT POINTER,
4468 025052 113737 002451 002450 29$:  MOVB     TMP4+1, TMP4 ;USE "NEXT" DATA AS "CURRENT" DATA
4469 025060 013700 002470    MOV      TMPC, RO    ;USE ONE BIT OF THE "DATA" SWITCH TO
4470 025064 006000          ROR      RO
4471 025066 106137 002451    ROLB     TMP4+1      ;BUILD A NEW "NEXT" DATA VALUE
4472 025072 013737 002474 002464    MOV      TMPE, TMPA  ;
4473 025100 000137 024362    58$:  JMP      10$          ;      POINT TO LAST ADDRESS AGAIN,
4474                                     ;      & TEST IT
4475          ;          "BKWD" SEQUENCE "DATA = 1?"
4476
4477 025104 000240          30$:  NOP
4478 025106 005137 002470    COM      TMPC        ;SWITCH DATA TYPE
4479 025112 001003          BNE      32$          ;NOW 1'S -- CHECK ADDRESS'S "LSB"
4480 025114 005037 002466    CLR      TMPC        ;NOW 0'S -- POINT TO BIT POSITION 0 AGAIN
4481 025120 000754          BR      29$          ;      RESET ADDRESS & TEST IT
4482
4483
4484          ; --- "STOP"
4485
4486 025122 000240          32$:  NOP
4487 025124 004537 004322    38$:  JSR      R5, WRITEI ;CLEAR RAM LOCATION 00B3 (HEX) & EXIT
4488 025130 000173          173          ;      (THIS CONVERTS TO 00B3 HEX.)
4489 025132 000000          0            ;      (THIS WE HOPE, WILL CLEAR IT)
4490 025134 103001          BCC      .+4         ;IF NO ERROR, PROCEED
4491 025136          ERROR          ;ELSE, REPORT IT
4492 025140 104460          ENDTST          ;THATS ALL FOLKS!
4493 025140          ;          TRAP      C$ERROR
4494 025140 104401          L10040:          TRAP      C$ETST
;.....
.EVEN

```


TEST 11 VIA REGISTER ADDRESSING

4520

.SBTTL TEST 11 - VIA REGISTER ADDRESSING

```

;*****
;
; TEST 11 -- VIA REGISTER ADDRESSING
;
; VIA == "6522 VERSATILE INTERFACE ADAPTER"
;
; A MASTER CLEAR IS PERFORMED, NEXT, TIMER 1 LATCHES
; ARE CLEARED BY WRITING 000 INTO VIA REGS 6 & 7
; THEN, 377 IS LOADED INTO DATA DIRECTION REGISTERS A, B (DDRA, DDRB) TO
; SET THE PORT PINS FOR OUTPUT MODE.
; THEN, A DIFFERENT BYTE OF DATA PATTERN C IS WRITTEN INTO EACH VIA
; LOCATION, (EXCEPT THE TIMER REGS 4,5,10,11 OCT) AND AFTER EACH IS WRITTEN,
; ALL VIA REGS (EXCEPT 4,5,10,11) ARE READ AND COMPARED TO EXPECTED
; CONTENTS. NOTE THAT SOME VIA REGS ARE ALTERED BY THE LOADING OF OTHERS,
; AND THE PROGRAM TAKES THIS INTO ACCOUNT, IN THE SETTING OF EXPECTED REG
; VALUES. THE DATA PATTERN IS CHOSEN TO AVOID ACTIVATING THE VIA CHIP (SUCH
; AS GENERATING OUTPUTS ON CA1, CA2, CB1, CB2, OR CAUSING 6502
; INTERRUPT REQUESTS).
; DATA PATTERN C (WITH VIA REGS AND THEIR DATA SHOWN IN OCTAL) :
; REGISTER = 00 01 02 03 06 07 12 13 14 15 16 17
; DATA = 100, 101, 377, 377, 106, 107, 112, 040, 042, 000, 200, 117
; NEXT, 000 IS LOADED INTO DDRA, AND DDRB IS READ AND COMPARED TO 377. THEN,
; THE 377 IS LOADED BACK INTO DDRA, AND DDRB IS LOADED WITH 000 AND DDRA IS
; READ AND COMPARED TO 377.
;
;*****

```

```

;
; BGNTST
;
; T11::
4521 025142 004737 003762 JSR PC,MSTCLR ;INIT DMV AND START UP THE MAINT. LOOP
4522 025146 103002 BCC 1$ ;IF NO ERROR, PROCEED
4523 025150 10446C ERROR ;ELSE, REPORT IT AND
4524 025152 000546 BR 25$ ; EXIT THIS CLEAR TRAP C$ERROR
4525
4526 025154 004537 004322 1$: JSR R5,WRITEI ;CLEAR THE TIMER 1 LATCHES
4527 025160 120006 TILL
4528 025162 000000 0
4529 025164 103002 BCC 30$ ;IF AN ERROR OCCURED,
4530 025166 10446C ERROR ;REPORT IT &
4531 025170 000537 004322 30$: BR 25$ ; EXIT TRAP C$ERROR
4532 025172 004537 004322 JSR R5,WRITEI
4533 025176 120007 TILH
4534 025200 000000 0
4535 025202 103002 BCC 31$ ;IF AN ERROR OCCURED,
4536 025204 10446C ERROR ;REPORT IT &
4537 025206 000530 BR 25$ ; EXIT TRAP C$ERROR
4538
4539 ; LOAD UP THE VIA'S REGISTERS WITH THE FIXED DATA STREAM OF PATTERN 'C'
4540
4541 025210 013703 002556 31$: MOV PATC,R3 ;GET COUNT OF # OF WRITES TO BE PERFORMED
4542 025214 012702 002560 MOV @PATC+2,R2 ;SETUP POINTER TO REGISTER ADDRESSES & DATA
4543

```

TEST 11 VIA REGISTER ADDRESSING

```

4544 025220 012737 120000 025242 2$:  MOV   #ORB,4$ ;ADDRESS OF FIRST REGISTER
4545 025226 152237 025242          BISB  (R2)+,4$ ;OR IN REGISTER # TO BUILD REGISTER ADDRESS
4546 025232 112237 025244          MOVB  (R2)+,5$ ;THIS IS THE DATA WE WANT TO WRITE
4547
4548 025236 004537 004322          JSR   R5,WRITEI ;WRITE ONE REGISTER WITH THE DESIRED DATA
4549 025242 000000          4$:   0 ;*** MODIFIED FROM ABOVE *** DESTINATION ADDR.
4550 025244 000000          5$:   0 ;*** MODIFIED FROM ABOVE *** DATA
4551
4552 025246 103002          BCC   32$ ;IF AN ERROR OCCURED,
4553 025250          ERROR ;REPORT IT &
4554 025252 000506          BR    25$ ; EXIT TRAP C$ERROR
4555 025254 077317          32$: SOB  R3,2$ ;LOOP UNTIL THE WHOLE TABLE HAS BEEN WRITTEN
4556
4557 ; READ BACK THE VIA'S REGISTERS
4558
4559 025256 012703 000020          MOV   #PATCM PATCR,R3 ;GET COUNT OF # OF REG'S TO BE READ
4560 025262 012737 120000 025302          MOV   #ORB,7$ ;ADDRESS OF FIRST REGISTER
4561 025270 012737 003122 025304          MOV   #BT1,8$ ;DESTINATION BUFFER AREA
4562
4563 025276 004537 004064          6$:   JSR   R5,READ ;READ ONE REGISTER
4564 025302 000000          7$:   0 ;*** MODIFIED FROM ABOVE *** SOURCE ADDRESS
4565 025304 000000          8$:   0 ;*** MODIFIED IN LINE *** DESTINATION ADDRESS
4566 025306 103002          BCC   33$ ;IF AN ERROR OCCURED,
4567 025310          ERROR ;REPORT IT &
4568 025312 000466          BR    25$ ; EXIT TRAP C$ERROR
4569
4570 025314 005237 025302          33$: INC  7$ ;POINT TO NEXT REGISTER
4571 025320 005237 025304          INC  8$ ;POINT TO NEXT BUFFER LOCATION
4572 025324 077314          SOB  R3,6$ ;LOOP UNTIL ALL REGISTERS HAVE BEEN READ
4573
4574 ; CHECK THE VALUES READ AGAINST THE EXPECTED VALUES
4575
4576 025326 012701 002604          MOV   #PATCR,R1 ;POINTER TO EXPECTED DATA VALUES
4577 025332 012702 003122          MOV   #BT1,R2 ;POINTER TO DATA READ
4578 025336 012704 003206          MOV   #BT2,R4 ;POINTER TO "XOR" VALUES
4579 025342 012705 002624          MOV   #PATCM,R5 ;POINTER TO "MASK" VALUES
4580 025346 012703 000010          MOV   #8.,R3 ;NUMBER OF WORDS TO BE PROCESSED
4581 025352 005037 002332          CLR  ERRFLG ;RESET THE ERROR FLAG
4582
4583 025356 012114          9$:   MOV   (R1)+,(R4) ;GET EXPECTED VALUE (2 BYTES AT A TIME)
4584 025360 012200          MOV   (R2)+,R0 ;GET ACTUAL VALUE AND SETUP FOR "XOR"
4585 025362 074014          XOR  R0,(R4) ;DEVELOPE "XOR"
4586 025364 042524          BIC  (R5)+,(R4)+ ;CLEAR THOSE BITS WE DON'T CARE ABOUT
4587 025366 001402          BEQ  10$ ;IF NO ERROR, SKIP NEXT INSTRUCTION
4588 025370 005237 002332          INC  ERRFLG ;IF ERROR, SET FLAG TO SAY SO!
4589 025374 077310          10$: SOB  R3,9$ ;LOOP UNTIL ALL VALUES CHECKED
4590
4591 025376 005737 002332          TST  ERRFLG ;WAS THERE AN ERROR DETECTED?
4592 025402 001406          BEQ  12$ ;NO, PROCEED WITH TESTING
4593 025404          GEDF EM20,ERR6 ;YES, REPORT A VIA REGISTER ERROR
; "DEVICE FATAL" ERROR # 27
025404 104455          TRAP C$ERDF
025406 000033          .WORD 27
025410 015243          .WORD EM20
025412 005650          .WORD ERR6

```

TEST 11 VIA REGISTER ADDRESSING

```

4594 025414          ESCAPE TST          ;EXIT FROM THIS TEST -- LOOP IF REQUESTED
      025414 104410          TRAP      C$ESCAPE
      025416 000344          .WORD    L10041-.
4595
4596
4597
4598 025420 004537 004176 12$: JSR      R5,READI          ;GET THE CURRENT VALUE OF THE VIA'S
4599 025424 120003          DDRA          ; "DDRA" REGISTER FOR LATER ERROR CHECKING
4600 025426 000000          15$: 0
4601 025430 103002          BCC      34$          ;IF AN ERROR OCCURED,
4602 025432          ERROR          ;REPORT IT &
      025432 104460          TRAP      C$ERROR
4603 025434 000415          BR       25$
4604 025436 004537 004322 34$: JSR      R5,WRITEI          ;LOAD DDRB WITH 000
4605 025442 120002          DDRB
4606 025444 000000          0
4607 025446 103002          BCC      35$          ;IF AN ERROR OCCURED,
4608 025450          ERROR          ;REPORT IT &
      025450 104460          TRAP      C$ERROR
4609 025452 000406          BR       25$
4610 025454 004537 004064 35$: JSR      R5,READ          ; READ IT BACK AND CHECK IT
4611 025460 120002          DDRB
4612 025462 002312          BDATA
4613 025464 103002          BCC      36$          ;IF AN ERROR OCCURED,
4614 025466          ERROR          ;REPORT IT &
      025466 104460          TRAP      C$ERROR
4615 025470 000534          BR       24$
4616 025472 105737 002312 36$: TSTB   BDATA          ;THIS SHOULD NOW BE ZERO
4617 025476 001413          BEQ     14$          ;IT IS, PRECEDE TESTING
4618 025500 105037 002310 CLRB   GDATA          ;IT ISN'T! SETUP FOR & REPORT ERROR
4619 025504 012737 000002 002334 MOV    #2,REGNUM          ;IDENTIFY THE DDRB REG.
4620 025512          GEDF   EM21,ERR7          ; REPORT ERROR
      025512 104455          ; "DEVICE FATAL" ERROR # 28
      025514 000034          TRAP      C$ERDF
      025516 015442          .WORD    28
      025520 006612          .WORD    EM21
      025522          .WORD    ERR7
4621 025522          ESCAPE TST          ;EXIT FROM THIS TEST -- LOOP IF REQUESTED
      025522 104410          TRAP      C$ESCAPE
      025524 000236          .WORD    L10041 .
4622
4623 025526 113737 025426 002310 14$: MOVB   15$,GDATA          ;THIS IS WHAT WE EXPECT TO READ NOW
4624 025534 004537 004064 JSR      R5,READ          ;READ BACK DDRA - IT SHOULD BE = 366
4625 025540 120003          DDRA
4626 025542 002312          BDATA
4627 025544 103002          BCC      37$          ;IF AN ERROR OCCURED,
4628 025546          ERROR          ;REPORT IT &
      025546 104460          TRAP      C$ERROR
4629 025550 000504          BR       24$
4630 025552 123737 002310 002312 37$: CMPB   GDATA,BDATA          ;IS IT REALLY A 377?
4631 025560 001411          BEQ     16$          ;YES, PROCEED WITH TESTING
4632 025562 012737 000003 002334 MOV    #3,REGNUM          ;IDENTIFY THE DDRA REG.
4633 025570          GEDF   EM22,ERR7          ;NO, REPORT ERROR
      025570 104455          ; "DEVICE FATAL" ERROR # 29
      025572 000035          TRAP      C$ERDF
      025574 015477          .WORD    29
      .WORD    EM22

```

TEST 11 VIA REGISTER ADDRESSING

```

025576 006612
4634 025600          ESCAPE TST          ;EXIT FROM THIS TEST  LOOP IF .WORD  ERR7
025600 104410          ;                               TRAP    C$ESCAPE
025602 000160          ;                               .WORD   L10041-.
4635
4636 025604 004537 004322 16$: JSR    R5,WRITEI  ;RE LOAD DDRB WITH 377
4637 025610 120002          DDRB
4638 025612 177777          17$:   1
4639 025614 103002          BCC    38$          ;IF AN ERROR OCCURED,
4640 025616          ERROR          ;REPORT IT &
025616 104460          ;                               TRAP    C$ERROR
4641 025620 000460          BR     24$          ; EXIT
4642 025622 004537 004322 38$: JSR    R5,WRITEI  ;AND NOW CLEAR DDRA TO ZEROS
4643 025626 120003          DDRA
4644 025630 000000          0
4645 025632 103002          BCC    39$          ;IF AN ERROR OCCURED,
4646 025634          ERROR          ;REPORT IT &
025634 104460          ;                               TRAP    C$ERROR
4647 025636 000451          BR     24$          ; EXIT
4648
4649 025640 004537 004064 39$: JSR    R5,READ    ;NOW, DID DDRA GO TO ZEROES
4650 025644 120003          DDRA
4651 025646 002312          BDATA
4652 025650 105737 002312          TSTB  BDATA
4653 025654 001413          BEQ   18$          ;YES, BUT WHAT ABOUT DDRB?
4654 025656 105037 002310          CLRB  GDATA
4655 025662 012737 000003 002334          MOV   #3,REGNUM
4656 025670          GEDF  EM21,ERR7          ;NO, SETUP FOR AND
; IDENTIFY THE DDRA REG.
; REPORT THE ERROR
; "DEVICE FATAL' ERROR # 30
;                               TRAP    C$ERDF
;                               .WORD   30
;                               .WORD   EM21
;                               .WORD   ERR7
025670 104455          ;                               .WORD   ERR7
025672 000036          ;                               TRAP    C$ESCAPE
025674 015442          ;                               .WORD   L10041 .
025676 006612          ;                               .WORD   L10041 .
4657 025700          ESCAPE TST          ;EXIT FROM THIS TEST  LOOP IF REQUESTED
025700 104410          ;                               TRAP    C$ESCAPE
025702 000060          ;                               .WORD   L10041 .
4658
4659 025704 004537 004064 18$: JSR    R5,READ    ;WHAT ABOUT DDRB -- IT SHOULD BE 377 NOW
4660 025710 120002          DDRB
4661 025712 002312          BDATA
4662 025714 103002          BCC    40$          ;IF AN ERROR OCCURED,
4663 025716          ERROR          ;REPORT IT &
025716 104460          ;                               TRAP    C$ERROR
4664 025720 000420          BR     24$          ; EXIT
4665 025722 123737 002312 025612 40$: CMPB  BDATA,17$
4666 025730 001414          BEQ   24$          ;IS IT?
4667 025732 113737 025612 002310          MOVB  17$,GDATA
4668 025740 012737 000002 002334          MOV   #2,REGNUM
4669 025746          GEDF  EM22A,ERR7          ;YES, EXIT TEST
;NO, SETUP FOR AND
; IDENTIFY THE DDRB REG.
; REPORT ERROR
; "DEVICE FATAL' ERROR # 31
;                               TRAP    C$ERDF
;                               .WORD   31
;                               .WORD   EM22A
;                               .WORD   ERR7
025746 104455          ;                               .WORD   ERR7
025750 000037          ;                               TRAP    C$ESCAPE
025752 015532          ;                               .WORD   L10041 .
025754 006612          ;                               .WORD   L10041 .
4670 025756          ESCAPE TST          ;EXIT FROM THIS TEST  LOOP IF REQUESTED
025756 104410          ;                               TRAP    C$ESCAPE
025760 000002          ;                               .WORD   L10041 .

```

H10

TEST 11 VIA REGISTER ADDRESSING

4671
4672 025762
 025762
 025762 104401

244: ENDTST

L10041: TRAP C#ETST

TEST 12 -- VIA'S DDRB DATA READ/WRITE

4684

.SBTTL TEST 12 -- VIA'S DDRB DATA READ/WRITE

```

*****
;*
;* TEST 12 - VIA'S DDRB DATA READ/WRITE
;*
;* DDRB == "DATA DIRECTION REGISTER B"
;* FIRST, A MASTER CLEAR IS PERFORMED. THEN :
;* READ/WRITE BITS 0-7 OF VIA DATA DIRECTION REGISTER B ARE TESTED BY WRITING,
;* READING, AND COMPARING EACH BYTE OF DATA PATTERN B.
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
;*
*****

```

```

:
: BGNTST
:
4685 025764 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
4686 025770 103003 BCC 30$ ;IF AN ERROR OCCURED.
4687 025772 ERROR ;REPORT IT &
4688 025774 104460 ESCAPE TST ; EXIT TRAP C$ERROR
025776 104410 ; TRAP C$ESCAPE
025776 000046 ;.WORD L10042-.
4689
4690 026000 012701 002526 30$: MOV #PATB,R1 ;POINT TO PATTERN TABLE
4691 026004 012103 MOV (R1)+,R3 ;GET # OF ENTRIES IN TABLE
4692
4693 026006 T12.LP: BGNSUB ;THE SUBTEST ONLY TESTS THE ONE PATTERN
4694 026006 ; T12.1: TRAP C$BSUB
026006 104402
4695
4696 026010 111137 002306 MOVB (R1),TDATA ;SETUP TEST DATA BYTE FOR "STREG"
4697 026014 112137 002310 MOVB (R1)+,GDATA ;SETUP EXPECTED DATA BYTE FOR "STREG"
4698 026020 012700 120002 MOV #DDR,B,R0 ;SPECIFY THE REGISTER BEING TESTED
4699 026024 004737 005034 JSR PC,STREG ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
4700 026030 103003 BCC 10$ ;WAS AN ERROR FOUND?
4701 026032 ERROR ;YES, REPORT IT AND.
4702 026034 104460 ESCAPE TST ; EXIT FROM THE TEST. "CKLOOP" IS IMPLIED TRAP C$ERROR
026036 000006 ; TRAP C$ESCAPE
; .WORD L10042.
4703
4704 026040 104403 10$: ENDSUB ; L10043: TRAP C$ESUB
026040
4705
4706 026042 077317 SOB R3,T12.LP ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
4707 ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST
4708
4709 026044 ENDTST
026044 ; L10042: TRAP C$ETST
026044 104401

```

TEST 13 VIA'S DDRA DATA READ/WRITE

4722

.SBTTL TEST 13 -- VIA'S DDRA DATA READ/WRITE

```

:*****
:
:  TEST 13 -- VIA'S DDRA DATA READ/WRITE
:
:  DDRA == "DATA DIRECTION REGISTER A"
:
:  FIRST, A MASTER CLEAR IS PERFORMED. THEN :
:  READ/WRITE BITS 0-7 OF VIA DATA DIRECTION REGISTER A ARE TESTED BY WRITING,
:  READING, AND COMPARING EACH BYTE OF DATA PATTERN B.
:  DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
:                    200, 376, 375, 373, 367, 357, 337, 277, 177, 000
:*****

```

```

:
:  BGNTST
:
:  JSR    PC,MSTCLR      ;INIT DMV & START UP THE MAINT. LOOP
:  BCC    30$           ;IF AN ERROR OCCURED,
:  ERROR                                ;REPORT IT &
:
:  ESCAPE TST           ; EXIT
:
:
:  TRAP   C$ERROR
:
:  ESCAPE TST           ; EXIT
:
:  TRAP   C$ESCAPE
:  .WORD  L10044 .
:
:  30$:  MOV    #PATB,R1 ;POINT TO PATTERN TABLE
:        MOV    (R1)+,R3 ;GET # OF ENTRIES IN TABLE
:
:  T13.LP:
:  BGNSUB                                ;THE SUBTEST ONLY TESTS THE ONE PATTERN
:
:  TRAP   C$BSUB
:
:  MOVB   (R1),TDATA  ;SETUP TEST DATA BYTE FOR "STREG"
:  MOVB   (R1)+,GDATA ;SETUP EXPECTED DATA BYTE FOR "STREG"
:  MOV    #DDRA,R0    ;SPECIFY THE REGISTER BEING TESTED
:  JSR    PC,STREG    ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
:  BCC    10$         ;WAS AN ERROR FOUND?
:  ERROR                                ;YES, REPORT IT AND
:
:  TRAP   C$ERROR
:  ESCAPE TST           ; EXIT FROM THE TEST. "CKLOOP" IS IMPLIED
:
:  TRAP   C$ESCAPE
:  .WORD  L10044 .
:
:  10$:  ENDSUB
:
:  L10045:
:  TRAP   C$ESUB
:
:  SOB    R3,T13.LP   ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
:                    ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST
:
:  ENDTST
:
:  L10044:
:  TRAP   C$ETST

```

```

026046
4723 026046 004737 003762
4724 026052 103003
4725 026054
4726 026054 104460
026056
026056 104410
026060 000046
4727
4728 026062 012701 002526
4729 026066 012103
4730
4731 026070
4732 026070
026070
026070 104402
4733
4734 026072 111137 002306
4735 026076 112137 002310
4736 026102 012700 120003
4737 026106 004737 005034
4738 026112 103003
4739 026114
026114 104460
4740 026116
026116 104410
026120 000006
4741
4742 026122
026122
026122 104403
4743
4744 026124 077317
4745
4746
4747 026126
026126
026126 104401

```

TEST 14 VIA'S ORB DATA READ/WRITE

4761

.SBTTL TEST 14 -- VIA'S ORB DATA READ/WRITE

```

*****
;*
;* TEST 14 -- VIA'S ORB DATA READ/WRITE
;*
;* ORB == "OUTPUT REGISTER PORT B"
;*
;* FIRST, A MASTER CLEAR IS PERFORMED. NEXT, 377 IS LOADED INTO DATA
;* DIR. REG. B (DDRB) TO SET ALL B PORT PINS FOR OUTPUT MODE. THEN
;* READ/WRITE BITS 0-7 OF VIA OUTPUT REG. PORT B ARE TESTED BY WRITING,
;* READING, AND COMPARING EACH BYTE OF DATA PATTERN B.
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
;*
*****

```

```

;
; BGNTS1
;
; T14::
4762 026130 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
4763 026134 103003 BCC 30$ ;IF AN ERROR OCCURED,
4764 026136 ERROR ;REPORT IT &
; TRAP C$ERROR
4765 026140 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
026140 104410 .WORD L10046-.
026142 000066
4766
4767 026144 004537 004322 30$: JSR R5,WRITEI ;INITIALIZE PORT B FOR I/O
4768 026150 120002 DDRB
4769 026152 177777 -1
4770 026154 103003 BCC 31$ ;IF AN ERROR OCCURED,
4771 026156 ERROR ;REPORT IT &
; TRAP C$ERROR
4772 026160 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
026160 104410 .WORD L10046-.
026162 000046
4773
4774 026164 012701 002526 31$: MOV #PATB,R1 ;POINT TO PATTERN TABLE
4775 026170 012103 MOV (R1)+,R3 ;GET # OF ENTRIES IN TABLE
4776
4777 026172 T14.LP: BGNSUB ;THE SUBTEST ONLY TESTS THE ONE PATTERN
4778 026172 ; T14.1:
026172 104402 ; TRAP C$BSUB
4779
4780 026174 111137 002306 MOVB (R1),TDATA ;SETUP TEST DATA BYTE FOR "STREG"
4781 026200 112137 002310 MOVB (R1)+,GDATA ;SETUP EXPECTED DATA BYTE FOR "STREG"
4782 026204 012700 120000 MOV #ORB,R0 ;SPECIFY THE REGISTER BEING TESTED
4783 026210 004737 005034 JSR PC,STREG ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
4784 026214 103003 BCC 10$ ;WAS AN ERROR FOUND?
4785 026216 ERROR ;YES, REPORT IT AND
; TRAP C$ERROR
026216 104460 ; EXIT FROM THE TEST. "CKLOOP" IS IMPLIED
4786 026220 ESCAPE TST ; TRAP C$ESCAPE
026220 104410 .WORD L10046-.
026222 000006
4787
4788 026224 10$: ENDSUB

```


TEST 14 VIA'S ORB DATA READ/WRITE

026224
 026224 104403
 4789
 4790 026226 077317
 4791
 4792
 4793 026230
 026230
 026230 104401

SOB R3,T14.LP
 ENDTST

;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
 ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST

L10047: TRAP C\$ESUB

L10046: TRAP C\$ETST

TEST 15 - VIA'S T1 DATA READ/WRITE

4854

.SBTTL TEST 15 -- VIA'S T1 DATA READ/WRITE

```

:*****
:
:* TEST 15 -- VIA'S T1 DATA READ/WRITE
:*
:* T1 == "TIMER #1"
:*
:* THIS TEST WRITES, READS, AND CHECKS THE T1 LATCHES AND COUNTER REGISTERS
:* WITH DATA PATTERNS IN EACH OF 3 SUBTESTS.
:*
:*
:* FIRST SUBTEST: CHECKS FOR PROPER LOADING OF LATCHES
:* IT ALSO CHECKS TO BE SURE THAT THE COUNTER APPEARS TO BE DOING
:* SOMETHING TO THE COUNTERS. AS LONG AS THEY HAVE CHANGED FROM THE
:* VALUE LOADED INTO THEM, WE WILL BE SATISFIED.
:*
:* A. A MASTER CLEAR IS PERFORMED.
:* B. ALL LATCHES ARE LOADED TO ZEROES (JUST IN CASE), ACR6 & ACR7 ARE SET
:* TO ZERO (MODE 00), AND "T1" INTERRUPT ENABLE FLAG IS CLEARED.
:*
:* C. T1L-L(ADR 04) IS LOADED WITH THE CURRENT BYTE OF DATA PATTERN B.
:* D. T1L-L(ADR 06) IS READ AND COMPARED TO THE BYTE JUST WRITTEN.
:* E. T1C-L(ADR 04) IS READ AND CHECKED TO BE DIFFERENT THAN THE TEST BYTE.
:*
:* F. T1L-L(ADR 06) IS LOADED WITH THE COMPLEMENT OF THE CURRENT DATA BYTE.
:* G. T1L-L(ADR 06) IS READ AND COMPARED TO THE BYTE JUST WRITTEN.
:*
:* H. T1L-L(ADR 06) IS RE-LOADED WITH 0 TO MAKE T1C-H DECREMENT FAST.
:* T1L-H(ADR 05) IS LOADED WITH THE ORIGINAL TEST DATA PATTERN BYTE.
:* I. T1L-H(ADR 07) IS READ AND COMPARED TO THE BYTE LOADED INTO T1L-H.
:*
:* J. T1C-H(ADR 05) IS READ AND CHECKED TO BE DIFFERENT THAN THE TEST BYTE.
:*
:* K. T1L-H(ADR 07) IS LOADED WITH THE COMPLEMENT OF THE CURRENT DATA BYTE.
:* L. T1L-H(ADR 07) IS READ AND COMPARED TO THE BYTE JUST LOADED.
:*
:* M. STEPS C-L ARE REPEATED USING EACH BYTE OF DATA PATTERN B.
:*
:*
:* SECOND SUBTEST: CHECKS FOR CROSS-TALK AND ADDRESSING ERRORS
:* FROM T1L-L TO T1L-H
:*
:* A. T1L-H(ADR 07) IS LOADED WITH 000 TO CLEAR IT.
:* B. T1L-L(ADR 06) IS LOADED WITH A BYTE OF DATA PATTERN B.
:* C. T1L-L(ADR 06) IS READ AND COMPARED TO THE DATA JUST WRITTEN.
:* D. T1L-H(ADR 07) IS READ AND COMPARED TO 000.
:* E. STEPS B-D ARE REPEATED USING EACH BYTE OF DATA PATTERN B.
:*
:*
:* THIRD SUBTEST: CHECKS FOR CROSS-TALK AND ADDRESSING ERRORS
:* FROM T1L-H TO T1L-L
:*
:* A. T1L-L(ADR 04) IS LOADED WITH 000 TO CLEAR IT
:* B. T1L-H(ADR 07) IS LOADED WITH A BYTE OF DATA PATTERN B.
:* C. T1L-H(ADR 07) IS READ AND COMPARED TO THE DATA JUST WRITTEN.
:* D. T1L-L(ADR 06) IS READ AND COMPARED TO 000.

```

```

;* E. STEPS B-D ARE REPEATED USING EACH BYTE OF DATA PATTERN B.
;*
;* DATA PATTERN B = 125, 252, 000, 377, 001, 002, 004, 010, 020, 040, 100,
;*                   200, 376, 375, 373, 367, 357, 337, 277, 177, 000
;*
;-----*

```

```

;
;       BGNTST
;
;                                     T15::
;
; ***** STEP A *****
;
;       JSR      PC,MSTCLR      ;INIT DMV & START UP M-LOOP
;       BCC     1$             ;IF NO ERRORS, PROCEED
;       ERRORR                               ;ELSE, REPORT ERROR &
;
;       ESCAPE  TST             ;       GET OUT OF THE TEST      TRAP    C$ERROR
;
;                                     TRAP    C$ESCAPE
;                                     .WORD   L10050-.
1$:

```

```

; ***** STEP B *****
;
;       JSR      R5,INITT1     ;INITIALIZE THE TIMER'S REGISTERS
;       O                               ; WITH ZEROES
;       .WORD   0               ; 00 --> ACR6 & ACR7 AND DISABLE INTERRUPTS
;       BCC     .+10           ;IF NO ERROR, PROCEED
;       ERROR                               ;ELSE, REPORT IT
;
;       ESCAPE  TST             ;       AND EXIT THIS TEST      TRAP    C$ERROR
;
;                                     TRAP    C$ESCAPE
;                                     .WORD   L10050-.

```

```

;WE WANT THE LEAST ACTIVE OPERATING MODE FOR THIS TIMER WHILE WE ARE TESTING
;IT. THE MODE WE'RE USING HERE IS DOCUMENTED THUSLY: "GENERATE A SINGLE
;TIME-OUT INTERRUPT EACH TIME T1 IS LOADED. PB7 DISABLED."
;AS AN ADDED PRECAUTION, WE ARE DISABLING INTERRUPTS BY CLEARING THE "T1" FLAG
;WITHIN "IER".

```

```

;
;       BGNSUB                               ;BEGIN THE FIRST SUBTEST
;
;                                     $T15.1:
;                                     TRAP    C$BSUB
;
;       MOV     #PATB,R1           ;POINT TO THE APPROPRIATE PATTERN TABLE
;       MOV     (R1)+,R3          ;EXTRACT THE BYTE COUNT FROM THE TABLE
;
;T16.LP:
;       MOVB   (R1)+,TDATA        ;GET ONE BYTE OF THE TEST DATA
;       MOV     TDATA,GDATA       ;THE TEST DATA IS NORMALLY THE GOOD DATA TOO
;
;       BGNSEG                               TRAP    C$BSEG

```

```

; ***** STEP C *****
;
;       JSR      R5,WRITE        ;LOAD T1L-L(ADDR 04)
;       T1CL
;       TDATA                               ;THE TEST DATA FROM "TDATA"

```

```

026232
4855
4856
4857
4858 026232 004737 003762
4859 026236 103003
4860 026240
4861 026242
4862 026246
4863
4864
4865
4866 026246 004537 004660
4867 026252 000000
4868 026254 000000
4869 026256 103003
4870 026260
4871 026262
4872
4873
4874
4875
4876
4877
4878
4879 026266
4880 026270 012701 002526
4881 026274 012103
4882
4883 026276
4884 026276 112137 002306
4885 026302 013737 002306 002310
4886
4887 026310
4888 026310 104404
4889
4890
4891 026312 004537 004310
4892 026316 120004
4893 026320 002306

```


TEST 15 VIA'S T1 DATA READ/WRITE

```

026506 000042
026510 015243
026512 006612
4939
4940
4941
4942 026514 105137 002306
4943 026520 105137 002310
4944 026524 004537 004322
4945 026530 120006
4946 026532 000001
4947 026534 004537 004310
4948 026540 120005
4949 026542 002306
4950
4951
4952
4953 026544 004537 004064
4954 026550 120007
4955 026552 002312
4956 026554 123737 002310 002312
4957 026562 001407
4958 026564 012737 000007 002334
4959 026572
026572 104455
026574 000043
026576 015243
026600 006612
4960
4961
4962
4963 026602 004537 004064
4964 026606 120005
4965 026610 002312
4966 026612 012737 000005 002334
4967 026620 105737 002306
4968 026624 001410
4969 026626 123737 002310 002312
4970 026634 001004
4971 026636
026636 104455
026640 000044
026642 015275
026644 006612
4972
4973
4974
4975 026646 105137 002306
4976 026652 105137 002310
4977 026656 004537 004310
4978 026662 120007
4979 026664 002306
4980
4981
4982

```

```

; ***** STEP H *****
8$: COMB TDATA ;RESTORE THE DATA TO THE ORIGINAL VALUE
COMB GDATA
JSR R5,WRITEI ;SET THE LOW LATCH TO MAKE SURE THE HIGH
TILL ;COUNTER IS DOING MOST OF THE WORK
1
JSR R5,WRITE ;LOAD T1L-H(ADDR 05)
T1CH
TDATA ;THE TEST DATA FROM "TDATA"

; *** - *** STEP I ***
4953 JSR R5,READ ;READ T1L-H(ADDR 07)
T1LH
BDATA
CMPB GDATA,BDATA ;AND CHECK IT
BEQ 10$ ;IF OK, PROCEED
MOV #7,REGNUM ;IDENTIFY THE FAILING REGISTER &
GEDF EM20,ERR7 ;REPORT FAILURE
; "DEVICE FATAL" ERROR # 35
TRAP C$ERDF
.WORD 35
.WORD EM20
.WORD ERR7

; **** - **** STEP J ****
10$: JSR R5,READ ;READ T1C H(ADDR 05)
T1CH
BDATA
MOV #5,REGNUM ;IDENTIFY THE REGISTER BEING CHECKED
TSTB TDATA ;WAS THE TEST DATA 000 ?
BEQ 14$ ;YES, THEN WE CAN'T BE SURE OF THE RESULTS!
CMPB GDATA,BDATA ;NO, CHECK IT
BNE 14$ ;IT SHOULDN'T = THE LOADED VALUE
GEDF EM20A,ERR7 ;IT DID! REPORT FAILURE
; "DEVICE FATAL" ERROR # 36
TRAP C$ERDF
.WORD 36
.WORD EM20A
.WORD ERR7

; AND CONTINUE TESTING

; ***** STEP K *****
14$: COMB TDATA ;USE THE ONE'S COMPLEMENT THIS TIME
COMB GDATA ;THE EXPECTED DATA IS ALSO THE COMPLEMENT
JSR R5,WRITE ;LOAD T1L-H(ADDR 07)
T1LH
TDATA ;THE TEST DATA FROM "TDATA"

; *** - *** STEP L ***

```

```

4983 026666 004537 004064      JSR      R5,READ      ;READ T1L H(ADDR 07)
4984 026672 120007
4985 026674 002312      BDATA
4986 026676 123737 002310 002312  CMPB     GDATA,BDATA ;AND CHECK IT
4987 026704 001407      BEQ      16$          ;IF OK, PROCEED
4988 026706 012737 000007 002334  MOV      #7,REGNUM   ;IDENTIFY THE FAILING REGISTER &
4989 026714      GEDF     EM20,ERR7   ; REPORT FAILURE
;          "DEVICE FATAL" ERROR # 37
;          TRAP      C$ERDF
;          .WORD    37
;          .WORD    EM20
;          .WORD    ERR7

      026714 104455
      026716 000045
      026720 015243
      026722 006612
4990
4991      ; *** - *** STEP M *** --***
4992
4993 026724      16$:      ENDSEG
      026724
      026724 104405      10000$:      TRAP      C$ESEG
4994
4995 026726 000402
4996 026730 000137 026276 20$:      BR      21$
4997 026734 077303 21$:      JMP      T16.LP
4998      SOB      R3,20$ ;IF MORE DATA, DO ANOTHER BYTE
4999 026736      ENDSUB   ;ELSE, EXIT SUBTEST
      026736
      026736 104403      L10051:     TRAP      C$ESUB
5000
5001
5002 026740      BGNSUB   ;BEGIN THE SECOND SUBTEST
      026740
      026740 104402      T15.2:     TRAP      C$BSUB
5003 026742 012701 002526  MOV      #PATB,R1   ;POINT TO THE APPROPRIATE PATTERN TABLE
5004 026746 012103  MOV      (R1),R3   ;EXTRACT THE BYTE COUNT FROM THE TABLE
5005
5006 026750      T16.L1:
5007 026750 112137 002306  MOVB     (R1),.TDATA ;GET ONE BYTE OF THE TEST DATA
5008 026754 013737 002306 002310  MOV      TDATA,GDATA ;THE TEST DATA IS NORMALLY THE GOOD DATA TOO
5009
5010
5011      ; ***---*** STEP A *** ***
5012
5013 026762 004537 004322      JSR      R5,WRITEI   ;CLEAR T1L-H(ADDR 07)
5014 026766 120007      T1LH
5015 026770 000000      0          ;THE TEST DATA FROM 'TDATA'
5016
5017 026772      BGNSEG
      026772 104404      TRAP      C$BSEG
5018
5019      ; *** - *** STEP B *** ***
5020
5021 026774 004537 004310      JSR      R5,WRITE   ;LOAD T1L L(ADDR 06)
5022 027000 120006      T1LL
5023 027002 002*76      TDATA      ;THE TEST DATA FROM 'TDATA'
5024
5025      ; *** -*** STEP C *** --- ***
5026
5027 027004 004537 004064      JSR      R5,READ    ;READ T1L -L(ADDR 06)

```

TEST 15 -- VIA'S T1 DATA READ/WRITE

```

5028 027010 120006          T1LL
5029 027012 002312          BDATA
5030 027014 123737 002310 002312  CMPB  GDATA,BDATA ;AND CHECK IT
5031 027022 001407          BEQ    2$           ;IF OK, PROCEED
5032 027024 012737 000006 002334  MOV   #6,REGNUM   ;IDENTIFY THE FAILING REGISTER &
5033 027032          GEDF  EM20,ERR7 ; REPORT FAILURE
; "DEVICE FATAL" ERROR # 38
; TRAP C$ERDF
; .WORD 38
; .WORD EM20
; .WORD ERR7
027032 104455
027034 000046
027036 015243
027040 006612
5034
5035 ; ****--*** STEP D ****--***
5036
5037 027042 004537 004064 2$: JSR  R5,READ ;READ T1L-H(ADDR 07)
5038 027046 120007          T1LH
5039 027050 002312          BDATA
5040 027052 105737 002312  TSTB  BDATA ;AND CHECK IT -- THIS SHOULD STILL BE ZERO
5041 027056 001411          BEQ    10$         ;IF OK, PROCEED
5042 027060 005037 002310  CLR   GDATA
5043 027064 012737 000007 002334  MOV   #7,REGNUM   ;IDENTIFY THE FAILING REGISTER &
5044 027072          GEDF  EM20B,ERR7 ; REPORT FAILURE
; "DEVICE FATAL" ERROR # 39
; TRAP C$ERDF
; .WORD 39
; .WORD EM20B
; .WORD ERR7
027072 104455
027074 000047
027076 015355
027100 006612
5045
5046 ; ****--*** STEP E ****--***
5047
5048 027102          10$: ENDFSEG ;
027102          10000$: TRAP C$ESEG
027102 104405
5049
5050 027104 000402          BR    21$
5051 027106 000137 026750 20$: JMP  T16.L1
5052 027112 077303          21$: SOB  R3,20$ ;IF MORE DATA, DO ANOTHER BYTE
5053 ;ELSE, EXIT SUBTEST
5054 027114          ENDSUB
;
; L10052: TRAP C$ESUB
027114 104403
5055
5056
5057 027116          BGNSUB ;BEGIN THE THIRD SUBTEST
027116          T15.3: TRAP C$BSUB
027116 104402
5058 027120 012701 002526  MOV   #PATB,R1 ;POINT TO THE APPROPRIATE PATTERN TABLE
5059 027124 012103          MOV   (R1),R3 ;EXTRACT THE BYTE COUNT FROM THE TABLE
5060
5061 027126          T16.L2:
5062 027126 112137 002306  MOVB  (R1),TDATA ;GET ONE BYTE OF THE TEST DATA
5063 027132 013737 002306 002310  MOV   TDATA,GDATA ;THE TEST DATA IS NORMALLY THE GOOD DATA TOO
5064
5065
5066 ; ****--*** STEP A ****--***
5067
5068 027140 004537 004322  JSR  R5,WRITEI ;CLEAR T1L L(ADDR 04)

```

TEST 15 VIA'S T1 DATA READ/WRITE

SEQ 0135

```

5069 027144 120004          T1CL
5070 027146 000000          0
5071
5072 027150          BGNSEG
      027150 104404
5073                                     TRAP      C#BSEG
5074                                     ; ****--*** STEP B *** --****
5075
5076 027152 004537 004310      JSR      R5,WRITE      ;LOAD T1L-M(ADDR 07)
5077 027156 120007          T1LH
5078 027160 002306          TDATA      ;THE TEST DATA FROM "TDATA"
5079
5080                                     ; ****--*** STEP C ***- -****
5081
5082 027162 004537 004064      JSR      R5,READ      ;READ T1L-M(ADDR 07)
5083 027166 120007          T1LH
5084 027170 002312          BDATA
5085 027172 123737 002310 002312  CMPB     GDATA,BDATA  ;AND CHECK IT
5086 027200 001407          BEQ      10$          ;IF OK, PROCEED
5087 027202 012737 000007 002334  MOV      #7,REGNUM    ;IDENTIFY THE FAILING REGISTER &
5088 027210          GEDF     EM20,ERR7  ; REPORT FAILURE
                                     ;          "DEVICE FATAL" ERROR # 40
      027210 104455
      027212 000050          TRAP      C$ERDF
      027214 015243          .WORD    40
      027216 006612          .WORD    EM20
                                     .WORD    ERR7
5089
5090                                     ; ****--*** STEP D ***- -****
5091
5092 027220 004537 004064 10$:  JSR      R5,READ      ;READ T1L L(ADDR 06)
5093 027224 120006          T1LL
5094 027226 002312          BDATA
5095 027230 105737 002312      TSTB     BDATA
5096 027234 001411          BEQ      2$          ;AND CHECK IT
5097 027236 005037 002310      CLR      GDATA
5098 027242 012737 000006 002334  MOV      #6,REGNUM    ;IDENTIFY THE FAILING REGISTER &
5099 027250          GEDF     EM20B,ERR7  ; REPORT FAILURE
                                     ;          "DEVICE FATAL" ERROR # 41
      027250 104455          TRAP      C$ERDF
      027252 000051          .WORD    41
      027254 015355          .WORD    EM20B
      027256 006612          .WORD    ERR7
5100
5101                                     ; ****--*** STEP E ***   ***
5102
5103 027260          2$:  ENDSEG
      027260
      027260 104405          10000$: TRAP      C$ESEG
5104
5105 027262 000402          BR      21$
5106 027264 000137 027126 20$:  JMP      T16.L2
5107 027270 077303          21$:  SOB     R3,20$
5108                                     ;IF MORE DATA, DO ANOTHER BYTE
5109 027272          ENDSUB      ;ELSE, EXIT SUBTEST
      027272
      027272 104403          L10053: TRAP      C$ESUB
5110

```


G11

CNDMAAO DMV11 MCTRL DIAG #1 MACRO M1200 22 FEB 84 15:22 PAGE 65-7

SEQ 0136

TEST 15 VIA'S T1 DATA READ/WRITE

5111 027274
 027274
 027274 104401

ENDTST

L10050: TRAP C\$ETST

TEST 18 VIA'S PCR DATA READ/WRITE

5200

.SBTTL TEST 18 -- VIA'S PCR DATA READ/WRITE

```

*****
;*
;* TEST 18 -- VIA'S PCR DATA READ/WRITE
;*
;* PCR == "PERIPHERAL CONTROL REGISTER"
;*
;* FIRST, A MASTER CLEAR IS PERFORMED. THEN :
;* READ/WRITE BITS 0-7 OF THE PCR REGISTER ARE TESTED BY WRITING, READING,
;* AND COMPARING EACH BYTE OF A SUBSET OF DATA PATTERN B.
;* DATA PATTERN B (SUBSET) = 125, 252, 000, 377, 001, 002, 004, 010, 020,
;*                               040, 100, 200.
;*
*****

```

```

;
; BGNTST
;
5201 027442 004737 003762          T18::
5202 027446 103003          JSR    PC,MSTCLR    ;INIT DMV & START UP THE MAINT. LOOP
5203 027450          BCC    30$          ;IF AN ERROR OCCURED,
                                ERROR          ;REPORT IT &
                                TRAP    C$ERROR
5204 027452 104460          ESCAPE TST          ; EXIT
                                TRAP    C$ESCAPE
                                .WORD  L10060-.
5205
5206 027456 012701 002530    30$:  MOV    #PATB+2,R1    ;POINT TO PATTERN TABLE
5207 027462 012703 002543    MOV    #PATB+15,R3    ;GET # OF ENTRIES IN TABLE
5208
5209 027466          T20.LP:
5210 027466          BGNSUB          ;THE SUBTEST ONLY TESTS THE ONE PATTERN
                                T18.1:
                                TRAP    C$BSUB
5211
5212 027470 111137 002306    MOVB   (R1),TDATA    ;SETUP TEST DATA BYTE FOR "STREG"
5213 027474 112137 002310    MOVB   (R1)+,GDATA  ;SETUP EXPECTED DATA BYTE FOR "STREG"
5214 027500 012700 120014    MOV    #PCR,R0      ;SPECIFY THE REGISTER BEING TESTED
5215 027504 004737 005034    JSR    PC,STREG     ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
5216 027510 103003          BCC    10$          ;WAS AN ERROR FOUND?
5217 027512          ERROR          ;YES, REPORT IT AND
                                TRAP    C$ERROR
5218 027514 104460          ESCAPE TST          ; EXIT FROM THE TEST. "CKLOOP" IS IMPLIED
                                TRAP    C$ESCAPE
                                .WORD  L10060-.
5219
5220 027520 104403          10$:  ENDSUB          L10061:
                                TRAP    C$ESUB
5221
5222 027522 077317          SOB    R3,T20.LP    ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
5223
5224 027524          ENDTST          ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST
                                L10060:
                                TRAP    C$ETST

```

TEST 19 VIA'S IER DATA READ/WRITE

5243

.SBTTL TEST 19 -- VIA'S IER DATA READ/WRITE

```

:*****
:
:  TEST 19 -- VIA'S IER DATA READ/WRITE
:
:  IER == "INTERRUPT ENABLE REGISTER"
:
:  BITS 0-6 OF THE IER CAN BE SET OR CLEARED ON A WRITE, UNDER CONTROL OF THE
:  SET/CLEAR CONTROL BIT 7. TO TEST THIS , EACH BYTE OF DATA PATTERN D IS
:  WRITTEN INTO IER, AND THE REGISTER IS READ AND COMPARED TO THE CORRESPOND-
:  ING BYTE OF DATA PATTERN E.
:
:  DATA PATTERN D = 200, 201, 202, 204, 210, 220, 240, 300, 200, 000, 001,
:                   002, 004, 010, 020, 040, 100, 000, 325, 125, 252, 052
:
:  DATA PATTERN E = 000, 001, 003, 007, 017, 037, 077, 177, 177, 177, 176,
:                   174, 170, 160, 140, 100, 000, 000, 125, 000, 052, 000
:
:*****

```

```

:
:  BGNTST
:
5244 027526 004737 003762      JSR    PC,MSTCLR      ;INIT DMV & START UP THE MAINT. LOOP
5245 027532 103003              BCC    30$            ;IF AN ERROR OCCURED,
5246 027534 104460              ERROR   ;REPORT IT &
:                                     TRAP    C$ERROR
5247 027536 104410              ESCAPE TST           ; EXIT
:                                     TRAP    C$ESCAPE
:                                     .WORD   L10062 .
5248
5249 027542 012701 002644      30$:  MOV    #PATD,R1   ;POINT TO PATTERN TABLE
5250 027546 012103              MOV    (R1)+,R3      ;GET # OF ENTRIES IN TABLE
5251 027550 012702 002676      MOV    #PATE+2,R2   ;POINT TO "EXPECTED" DATA PATTERN TABLE
5252
5253 027554              T21.LP:
5254 027554              BGNSUB              ;THE SUBTEST ONLY TESTS THE ONE PATTERN
:                                     T19.1:
:                                     TRAP    C$BSUB
5255
5256 027556 112137 002306      MOVB   (R1)+,TDATA   ;SETUP TEST DATA BYTE FOR "STREG"
5257 027562 112237 002310      MOVB   (R2)+,GDATA   ;SETUP EXPECTED DATA BYTE FOR "STREG"
5258 027566 012700 120016      MOV    #IENR,R0     ;SPECIFY THE REGISTER BEING TESTED
5259 027572 004737 005034      JSR    PC,STREG      ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
5260 027576 103003              BCC    10$            ;WAS AN ERROR FOUND?
5261 027600 104460              ERROR   ;YES, REPORT IT AND
:                                     TRAP    C$ERROR
5262 027602 104410              ESCAPE TST           ; EXIT FROM THE TEST. "CKLOOP" IS IMPLIED
:                                     TRAP    C$ESCAPE
:                                     .WORD   L10062 .
5263
5264 027606 104403              10$:  ENDSUB
:                                     L10063:
:                                     TRAP    C$ESUB
5265
5266 027610 077317              SOB    R3,T21.LP     ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO

```

L11

TEST 19 VIA'S IER DATA READ/WRITE

5267
5268
5269 027612
 027612
 027612 104401

ENDTST

;TEST IT. ELSE, FALL OUT OF LOOP AND TEST

L10062: TRAP C↓ETST

TEST 20 VIA'S ORB/DRB MASTER CLEAR TEST

5282

.SBTTL TEST 20 -- VIA'S ORB/DRB MASTER CLEAR TEST

```

;*****
;*
;* TEST 20 -- VIA'S ORB/DRB MASTER CLEAR TEST
;*
;* ORB == "OUTPL, REGISTER PORT B"
;* DDRB == "DATA DIRECTION REGISTER B"
;*
;* FIRST, A MASTER CLEAR IS PERFORMED. NEXT, 377 IS LOADED INTO DDRB TO SET
;* ALL B PORT PINS FOR OUTPUT MODE. THEN, A 000 BYTE IS WRITTEN INTO ORB AND
;* THE REGISTER IS READ BACK AND CHECKED FOR 000. THEN, A MASTER CLEAR IS
;* PERFORMED AND ORB IS READ AND CHECKED FOR 377.
;*
;--*****

```

```

; BGNTST
;
; T20::
5283 027614
5284 027614 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
5285 027620 103003 BCC 1$ ;IF AN ERROR OCCURED,
5286 027622 ERROR ;REPORT IT &
; TRAP C$ERROR
5287 027624 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
; .WORD L10064-.
5288 027626 000252
5289 027630 012737 000377 002310 1$: MOV #377,GDATA ;SETUP FOR CALL TO STREG
5290 027636 013737 002310 002306 MOV GDATA,TDATA
5291
5292 ; WE'LL USE "STREG" TO LOAD & CHECK "DRB" WITH 377 (HEREBY SETTING UP
5293 ; "ORB" FOR BI-DIRECTIONAL TRANSFERS
5294
5295 027644 012700 120002 MOV #DDR,R0 ;POINT TO DRB
5296 027650 004737 005034 JSR PC,STREG ;LOAD & TEST IT
5297 027654 103003 BCC 4$ ;IF OK, PROCEED WITH TESTING
5298 027656 ERROR ;ELSE, REPORT THE ERROR
; TRAP C$ERROR
5299 027660 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10064-.
5300
5301 ; NOW WE'LL USE "STREG" TO SET & CHECK "ORB"
5302
5303 027664 012700 120000 4$: MOV #ORB,R0 ;POINT TO DRB
5304 027670 004737 005034 JSR PC,STREG ;LOAD & TEST "ORB"
5305 027674 103003 BCC 5$ ;IF NO ERROR HERE, PROCEED
5306 027676 ERROR ;ELSE, REPORT THE ERROR
; TRAP C$ERROR
5307 027700 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10064-.
5308
5309 027704 004737 003762 5$: JSR PC,MSTCLR ;ISSUE THE MASTER CLEAR (STAY IN M LOOP)
5310 027710 103003 BCC 10$ ;IF NO ERROR HERE, PROCEED
5311 027712 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR

```

TEST 20 -- VIA'S ORB/DRB MASTER CLEAR TEST

SEQ 0143

```

5312 027714          ESCAPE TST          ;      & QUIT          TRAP      C$ESCAPE
      027714 104410          ;                                .WORD      L10064-.
      027716 000162          ;                                ;
5313
5314 027720 005037 002310 10$: CLR      GDATA          ;FOR TESTING PURPOSES LATER
5315
5316 027724 004537 004064      JSR      R5,READ          ;READ THE "RESET" VALUE OF THE "DRB"
5317 027730 120002          DDRB
5318 027732 002312          BDATA
5319 027734 103003          BCC      12$            ;IF NO ERROR READING IT, PROCEED
5320 027736          ERROR          ;ELSE, REPORT IT
      027736 104460          ;                                TRAP      C$ERROR
5321 027740          ESCAPE TST          ;      & QUIT          TRAP      C$ESCAPE
      027740 104410          ;                                .WORD      L10064-.
      027742 000136          ;                                ;
5322
5323 027744 123737 002312 002310 12$: CMPB   BDATA,GDATA      ;DID IT GET CLEARE?
5324 027752 001407          BEQ      14$            ;YES, GOOD. NOW CHECK "ORB"
5325 027754 012737 000002 002334      MOV      #DDRBE<17>,REGNUM ;NO! BUILD REGISTER # POINTER
5326 027762          GEDF     EMS,ERR7      ;REPORT MASTER CLEAR FAILURE
      ;                                ; "DEVICE FATAL" ERROR # 42
      ;                                TRAP      C$ERDF
      ;                                .WORD      42
      ;                                .WORD      EMS
      ;                                .WORD      ERR7
      027762 104455          ;
      027764 000052          ;
      027766 014515          ;
      027770 006612          ;
5327
5328 027772 012737 000377 002310 14$: MOV      #377,GDATA      ;SETUP FOR CALL TO STREG
5329 030000 013737 002310 002306      MOV      GDATA,TDATA
5330
5331          ; WE'LL USE "STREG" TO LOAD & CHECK "DRB" WITH 377 THEREBY SETTING UP
5332          ; "ORB" FOR BY-DIRECTIONAL TRANSFERS
5333
5334 030006 012700 120002          MOV      #DRB,R0          ;POINT TO ORB
5335 030012 004737 005034          JSR      PC,STREG        ;LOAD & TEST IT
5336 030016 103003          BCC      16$            ;IF OK, PROCEED WITH TESTING
5337 030020          ERROR          ;ELSF, REPORT THE ERROR
      030020 104460          ;                                TRAP      C$ERROR
5338 030022          ESCAPE TST          ;      & QUIT          TRAP      C$ESCAPE
      030022 104410          ;                                .WORD      L10064 .
      030024 000054          ;                                ;
5339
5340 030026 005037 002310 16$: CLR      GDATA          ;SETUP FOR TESTING ORB
5341 030032 004537 004064      JSR      R5,READ          ;NOW READ THE "RESET" VALUE OF "ORB"
5342 030036 120000          ORB
5343 030040 002312          BDATA
5344 030042 103003          BCC      18$            ;IF NO ERROR READING IT, PROCEED
5345 030044          ERROR          ;ELSE, REPORT IT
      030044 104460          ;                                TRAP      C$ERROR
5346 030046          ESCAPE TST          ;      & QUIT          TRAP      C$ESCAPE
      030046 104410          ;                                .WORD      L10064 .
      030050 000030          ;                                ;
5347
5348 030052 123737 002310 002312 18$: CMPB   GDATA,BDATA      ;WAS IT PROPERLY RESET?
5349 030060 001407          BEQ      32$            ;YES, THIS TEST IS DONE, EXIT
5350 030062 012737 000000 002334      MOV      #ORB<17>,REGNUM ;NO! BUILD REGISTER # POINTER
5351 030070          GEDF     EMS,ERR7      ;REPORT MASTER CLEAR FAILURE
      ;                                ; "DEVICE FATAL" ERROR # 43
      ;                                ;

```


TEST 20 VIA'S ORB/DDR8 MASTER CLEAR TEST

SEQ 0144

030070 104455
030072 000053
030074 014515
030076 006612

TRAP C\$ERDF
.WORD 43
.WORD EMS
.WORD ERR7

5352
5353

030100
030100
030100 104401

328: ENDTST

L10064:

TRAP C\$ETST

5367

.SBTTL TEST 21 -- VIA'S DDRB MASTER CLEAR TEST

```

;.....
;
; TEST 21 -- VIA'S DDRB MASTER CLEAR TEST
;
; DDRB -- "DATA DIRECTION REGISTER B"
;
; A 377 BYTE IS WRITTEN INTO DDRB AND THE REGISTER IS READ BACK AND CHECKED
; FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND DDRB IS READ AND CHECKED FOR
; 000.
;
; NOTE: THIS TESTING IS ALSO DONE IN TEST 23. IT IS INCLUDED HERE ONLY TO
; PROVIDE TIGHTER LOOPING ON JUST THE DDRB MASTER CLEAR CHECKING.
;
;.....
;
; BGNTS1
;
; T21::

```

```

5368 030102
5369 030102 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
5370 030106 103003 BCC 1$ ;IF AN ERROR OCCURED.
5371 030110 104460 ERROR ;REPORT IT &
5372 030112 ESCAPE TST ; EXIT TRAP C$ERROR
030112 104410 ; TRAP C$ESCAPE
030114 000114 .WORD L10065-.
5373
5374 030116 012737 000377 002310 1$: MOV #377,GDATA ;SETUP FOR CALL TO STREG
5375 030124 013737 002310 002306 MOV GDATA,TDATA
5376 030132 012700 120002 MOV #DDR,B,R0
5377
5378 ; NOW WE'LL USE "STREG" TO SET & CHECK DDRB
5379
5380 030136 004737 005034 JSR PC,STREG ;LOAD & TEST 'DDR'B
5381 030142 103003 BCC 5$ ;IF NO ERROR HERE, PROCEED
5382 030144 104460 ERROR ;ELSE, REPORT IT
5383 030146 ESCAPE TST ; & QUIT TRAP C$ERROR
030146 104410 ; TRAP C$ESCAPE
030150 000060 .WORD L10065 .
5384
5385 030152 004737 003762 5$: JSR PC,MSTCLR ;ISSUE THE MASTER CLEAR (STAY IN M-LOOP)
5386 030156 103003 BCC 10$ ;IF NO ERROR HERE, PROCEED
5387 030160 104460 ERROR ;ELSE, REPORT IT
5388 030162 ESCAPE TST ; & QUIT TRAP C$ERROR
030162 104410 ; TRAP C$ESCAPE
030164 000044 .WORD L10065 .
5389
5390 030166 005037 002310 10$: CLR GDATA ;FOR TESTING PURPOSES LATER
5391 030172 004537 004064 JSR R5,READ ;NOW READ THE 'RESET' VALUE OF 'DDR'B"
5392 030176 120002 DDRB
5393 030200 002312 BDATA
5394
5395 030202 123737 002310 002312 CMPB GDATA,BDATA ;WAS IT PROPERLY RESET?
5396 030210 001407 BEQ 32$ ;YES, THIS TEST IS DONE, EXIT

```

TEST 21 VIA'S DDRB MASTER CLEAR TEST

5397 030212 012737 000002 002334
5398 030220

MOV #DDRBE<17>,REGNUM ;NO! BUILD REGISTER # POINTER
GEDF EMS,ERR7 ;REPORT MASTER CLEAR FAILURE
; "DEVICE FATAL" ERROR # 44

030220 104455
030222 000054
030224 014515
030226 006612

TRAP C\$ERDF
.WORD 44
.WORD EMS
.WORD ERR7

5399
5400 030230
030230
030230 104401

32\$: ENDTST

L10065:
TRAP C\$ETST

TEST 22 - VIA'S DDRA MASTER CLEAR TEST

5411

.SBTTL TEST 22 - VIA'S DDRA MASTER CLEAR TEST

```

;*****
;*
;* TEST 22 -- VIA'S DDRA MASTER CLEAR TEST
;*
;* DDRA == "DATA DIRECTION REGISTER A"
;*
;* A 377 BYTE IS WRITTEN INTO DDRA AND THE REGISTER IS READ BACK AND CHECKED
;* FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND DDRA IS READ AND CHECKED FOR
;* 000.
;*
;*****

```

```

;
; BGNTST
;
; T22::
5412 030232
5413 030232 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
5414 030236 103003 BCC 1$ ;IF AN ERROR OCCURED.
5415 030240 ERROR ;REPORT IT &
; TRAP C$ERROR
5416 030240 104460 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
; .WORD L10066-.
5417 030242 104410
5418 030244 000114
5419 030246 012737 000377 002310 1$: MOV #377,GDATA ;SETUP FOR CALL TO STREG
5420 030254 013737 002310 002306 MOV GDATA,TDATA
5421 030262 012700 120003 MOV #DDRA,RO
5422 ; NOW WE'LL USE "STREG" TO SET & CHECK 'DDRA"
5423
5424 030266 004737 005034 JSR PC,STREG ;LOAD & TEST "DDRA"
5425 030272 103003 BCC 5$ ;IF NO ERROR HERE, PROCEED
5426 030274 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5427 030274 104460 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10066 .
5428 030302 004737 003762 5$: JSR PC,MSTCLR ;ISSUE THE MASTER CLEAR (STAY IN M-LOOP)
5429 030306 103003 BCC 10$ ;IF NO ERROR HERE, PROCEED
5430 030310 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5431 030310 104460 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10066-.
5432 030312 104410
5433 030314 000044
5434 030316 005037 002310 10$: CLR GDATA ;FOR TESTING PURPOSES LATER
5435 030322 004537 004064 JSR R5,READ ;NOW READ THE 'RESET" VALUE OF "DDRA"
5436 030326 120003 DDRA
5437 030330 002312 BDATA
5438
5439 030332 123737 002310 002312 CMPB GDATA,BDATA ;WAS IT PROPERLY RESET?
5440 030340 001407 BEQ 32$ ;YES, THIS TEST IS DONE, EXIT
5441 030342 012737 000003 002334 MOV #DDRA<17>,REGNUM ;NO! BUILD REGISTER # POINTER
5442 030350 GEDF EM5,ERR7 ;REPORT MASTER CLEAR FAILURE
; "DEVICE FATAL" ERROR # 45

```

TEST 22 VIA'S DDRA MASTER CLEAR TEST

SEQ 0148

030350 104455
030352 000055
030354 014515
030356 006612

5443

5444 030360
030360
030360 104401

32\$: ENDTST

TRAP C\$ERDF
.WORD 45
.WORD EM5
.WORD ERR7

L10066: TRAP C\$ETST

TEST 23 VIA'S SR MASTER CLEAR TEST

5455

.SBTTL TEST 23 - VIA'S SR MASTER CLEAR TEST

```

;*****
;*
;* TEST 23 - VIA'S SR MASTER CLEAR TEST
;*
;* SR == 'SHIFT REGISTER'
;*
;* A 123 BYTE IS WRITTEN INTO SR AND THE REGISTER IS READ BACK AND CHECKED
;* FOR 123. THEN, A MASTER CLEAR IS PERFORMED AND SR IS READ AND CHECKED FOR
;* NO CHANGE.
;*
;*****

```

```

;
; BGNTST
;
; T23::
5456 030362
5457 030362 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
5458 030366 103003 BCC 1$ ;IF AN ERROR OCCURED,
5459 030370 ERROR ;REPORT IT &
; TRAP C$ERROR
5460 030370 104460 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
; .WORD L10067-.
5461 030372 104410
5462 030374 000120
5461 030376 004537 004322 1$: JSR R5,WRITEI ;FORCE SR TO MODE 0
5462 030402 120013 ACR
5463 030404 000000 0
5464 030406 012737 000123 002310 MOV #123,GDATA ;SETUP FOR CALL TO STREG
5465 030414 013737 002310 002306 MOV GDATA,TDATA
5466 030422 012700 120012 MOV #SR,R0
5467
5468
5469 ; NOW WE'LL USE "STREG" TO SET & CHECK 'SR'
5470
5471 030426 004737 005034 JSR PC,STREG ;LOAD & TEST "SR"
5472 030432 103003 BCC 5$ ;IF NO ERROR HERE, PROCEED
5473 030434 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5474 030434 104460 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10067-.
5475 030436 104410
5476 030440 000054
5476 030442 004737 003762 5$: JSR PC,MSTCLR ;ISSUE THE MASTER CLEAR (STAY IN M-LOOP)
5477 030446 103003 BCC 10$ ;IF NO ERROR HERE, PROCEED
5478 030450 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5479 030450 104460 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10067-.
5480 030452 104410
5481 030454 000040
5481 030456 004537 004064 10$: JSR R5,READ ;NOW READ THE "RESET" VALUE OF "SR"
5482 030462 120012 SR ; (IT SHOULDN'T HAVE CHANGED)
5483 030464 002312 BDATA
5484
5485 030466 123737 002310 002312 CMPB GDATA,BDATA ;WAS IT PROPERLY RESET?
5486 030474 001407 BEQ 32$ ;YES, THIS TEST IS DONE, EXIT
5487 030476 012737 000012 002334 MOV #SR<17>,REGNUM ;NO! BUILD REGISTER # POINTER

```

H12

TEST 23 VIA'S SR MASTER CLEAR TEST

SEQ 0150

5488 030504

GEDF EMS,ERR7

;REPORT MASTER CLEAR FAILURE
; "DEVICE FATAL" ERROR # 46

030504 104455
030506 000056
030510 014515
030512 006612

TRAP C\$ERDF
.WORD 46
.WORD EMS
.WORD FRR7

5489

5490 030514

32\$: ENDTST

030514
030514 104401

L10067:

TRAP C\$ETST

TEST 24 VIA'S ACR MASTER CLEAR TEST

5501

.SBTTL TEST 24 -- VIA'S ACR MASTER CLEAR TEST

```

:*****
:
:  TEST 24 -- VIA'S ACR MASTER CLEAR TEST
:
:  ACR == "AUXILIARY CONTROL REGISTER"
:
:  A 252 BYTE IS WRITTEN INTO ACR AND THE REGISTER IS READ BACK AND CHECKED
:  FOR 252. THEN, A MASTER CLEAR IS PERFORMED AND ACR IS READ AND CHECKED FOR
:  000, TO VERIFY THAT IT IS CLEARED BY MASTER CLEAR.
:
:*****

```

```

:      BGNTST
:
:      T24::
:
:      JSR      PC,MSTCLR      ;INIT DMV & START UP THE MAINT. LOOP
:      BCC     1$             ;IF AN ERROR OCCURED,
:      ERROR   ;REPORT IT &
:
:      ESCAPE  TST             ; EXIT
:
:      TRAP    C$ERROR
:
:      TRAP    C$ESCAPE
:      .WORD  L10070-.
:
:      1$:      MOV      #252,GDATA      ;SETUP FOR CALL TO STREG
:      MOV     GDATA,TDATA
:      MOV     #ACR,RO
:
:      ; NOW WE'LL USE "STREG" TO SET & CHECK "ACR" .
:
:      JSR      PC,STREG      ;LOAD & TEST "ACR"
:      BCC     5$             ;IF NO ERROR HERE, PROCEED
:      ERROR   ;ELSE, REPORT IT
:
:      TRAP    C$ERROR
:
:      ESCAPE  TST             ; & QUIT
:
:      TRAP    C$ESCAPE
:      .WORD  L10070-.
:
:      5$:      JSR      PC,MSTCLR      ;ISSUE THE MASTER CLEAR (STAY IN M LOOP)
:      BCC     10$            ;IF NO ERROR HERE, PROCEED
:      ERROR   ;ELSE, REPORT IT
:
:      TRAP    C$ERROR
:
:      ESCAPE  TST             ; & QUIT
:
:      TRAP    C$ESCAPE
:      .WORD  L10070-.
:
:      10$:     CLR      GDATA          ;FOR TESTING PURPOSES LATER
:      JSR     R5,READ        ;NOW READ THE "RESET" VALUE OF "ACR"
:      ACR     BDATA
:
:      CMPB   GDATA,BDATA     ;WAS IT PROPERLY RESET?
:      BEQ    32$             ;YES, THIS TEST IS DONE, EXIT
:      MOV    #ACR<17>,REGNUM ;NO! BUILD REGISTER # POINTER
:      GEDF   EMS,ERR7        ;REPORT MASTER CLEAR FAILURE
:
:      ; "DEVICE FATAL" ERROR # 47

```

```

030516
5502
5503 030516 004737 003762
5504 030522 103003
5505 030524
      030524 104460
5506 030526
      030526 104410
      030530 000114
5507
5508 030532 012737 000252 002310 1$:
5509 030540 013737 002310 002306
5510 030546 012700 120013
5511
5512
5513
5514 030552 004737 005034
5515 030556 103003
5516 030560
      030560 104460
5517 030562
      030562 104410
      030564 000060
5518
5519 030566 004737 003762 5$:
5520 030572 103003
5521 030574
      030574 104460
5522 030576
      030576 104410
      030600 000044
5523
5524 030602 005037 002310 10$:
5525 030606 004537 004064
5526 030612 120013
5527 030614 002312
5528
5529 030616 123737 002310 002312
5530 030624 001407
5531 030626 012737 000013 002334
5532 030634

```


J12

TEST 24 VIA'S ACR MASTER CLEAR TEST

030634 104455
030636 000057
030640 014515
030642 006612

TRAP C\$ERDF
.WORD 47
.WORD EMS
.WORD ERR7

5533

5534 030644

32\$: ENDTST

030644
030644 104401

L10070:

TRAP C\$ETST

TEST 25 VIA'S PCR MASTER CLEAR TEST

5545

.SBTTL TEST 25 -- VIA'S PCR MASTER CLEAR TEST

```

;*****
;
; TEST 25 - VIA'S PCR MASTER CLEAR TEST
;
; PCR == "PERIPHERAL CONTROL REGISTER"
;
; A 377 BYTE IS WRITTEN INTO PCR AND THE REGISTER IS READ BACK AND CHECKED
; FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND PCR IS READ AND CHECKED FOR
; 000.
;
;-----*****

```

```

;
; BGNTST
;
; T25::
5546 030646
5547 030646 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
5548 030652 103003 BCC 1$ ;IF AN ERROR OCCURED,
5549 030654 ERROR ;REPORT IT &
; TRAP C$ERROR
5550 030656 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
; .WORD L10071-.
5551
5552 030662 012737 000377 002310 1$: MOV #377,GDATA ;SETUP FOR CALL TO STREG
5553 030670 013737 002310 002306 MOV GDATA,TDATA
5554 030676 012700 120014 MOV #PCR,R0
5555
5556 ; NOW WE'LL USE "STREG" TO SET & CHECK "PCR"
5557
5558 030702 004737 005034 JSR PC,STREG ;LOAD & TEST "PCR"
5559 030706 103003 BCC 5$ ;IF NO ERROR HERE, PROCEED
5560 030710 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5561 030712 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10071 .
5562
5563 030716 004737 003762 5$: JSR PC,MSTCLR ;ISSUE THE MASTER CLEAR (STAY IN M LOOP)
5564 030722 103003 BCC 10$ ;IF NO ERROR HERE, PROCEED
5565 030724 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5566 030726 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10071 .
5567
5568 030732 005037 002310 10$: CLR GDATA ;FOR TESTING PURPOSES LATER
5569 030736 004537 004064 JSR R5,READ ;NOW READ THE 'RESET' VALUE OF PCR
5570 030742 120014 PCR
5571 030744 002312 BDATA
5572
5573 030746 123737 002310 002312 CMPB GDATA,BDATA ;WAS IT PROPERLY RESET?
5574 030754 001407 BEQ 32$ ;YES, THIS TEST IS DONE, EXIT
5575 030756 012737 000014 002334 MOV #PCR<17>,REGNUM ;NO! BUILD REGISTER # POINTER
5576 030764 GEDF EM5,ERR7 ;REPORT MASTER CLEAR FAILURE
; "DEVICE FATAL" ERROR # 48

```

TEST 25 VIA'S PCR MASTER CLEAR TEST

SEQ 0154

030764	104455
030766	000060
030770	014515
030772	006612

TRAP	C\$ERDF
.WORD	48
.WORD	EM5
.WORD	ERR7

5577			
5578	030774	32\$:	ENDTST
	030774		
	030774		104401

L10071:	TRAP	C\$ETST
---------	------	---------

TEST 26 VIA'S IER MASTER CLEAR TEST

5589

.SBTTL TEST 26 -- VIA'S IER MASTER CLEAR TEST

```

;*****
;*
;* TEST 26 -- VIA'S IER MASTER CLEAR TEST
;*
;* IER == "INTERRUPT ENABLE REGISTER"
;*
;* A 377 BYTE IS WRITTEN INTO IER AND THE REGISTER IS READ BACK AND CHECKED
;* FOR 377. THEN, A MASTER CLEAR IS PERFORMED AND IER IS READ AND CHECKED FOR
;* 200.
;*
;*****

```

```

;
; BGNTST
;
; T26::
5590 030776
5591 030776 004737 003762 JSR PC,MSTCLR ;INIT DMV & START UP THE MAINT. LOOP
5592 031002 103003 BCC 1$ ;IF AN ERROR OCCURED,
5593 031004 104460 ERROR ;REPORT IT &
; TRAP C$ERROR
5594 031006 ESCAPE TST ; EXIT
; TRAP C$ESCAPE
; .WORD L10072-.
5595
5596 031012 105077 151334 1$: CLRB @BSELO ;MAKE SURE NO Q-BUS INTERRUPTS RESULT FROM
5597 ; TESTING THE IER REGISTER
5598 031016 012737 000377 002310 MOV #377,GDATA ;SETUP FOR CALL TO STREG
5599 031024 013737 002310 002306 MOV GDATA,TDATA
5600 031032 012700 120016 MOV #IENR,R0
5601
5602 ; NOW WE'LL USE "STREG" TO SET & CHECK 'IER"
5603
5604 031036 004737 005034 JSR PC,STREG ;LOAD & TEST "IER"
5605 031042 103003 BCC 5$ ;IF NO ERROR HERE, PROCEED
5606 031044 104460 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5607 031046 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10072-.
5608
5609 031052 004737 003762 5$: JSR PC,MSTCLR ;ISSUE THE MASTER CLEAR (STAY IN M-LOOP)
5610 031056 103003 BCC 10$ ;IF NO ERROR HERE, PROCEED
5611 031060 104460 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR
5612 031062 ESCAPE TST ; & QUIT
; TRAP C$ESCAPE
; .WORD L10072 .
5613
5614 031066 012737 000200 002310 10$: MOV #200,GDATA ;FOR TESTING PURPOSES LATER
5615 031074 004537 004064 JSR R5,READ ;NOW READ THE "RESET" VALUE OF "IER"
5616 031100 120016 IENR
5617 031102 002312 BDATA
5618
5619 031104 123737 002310 002312 CMPB GDATA,BDATA ;WAS IT PROPERLY RESET?
5620 031112 001407 REQ 32$ ;YES, THIS TEST IS DONE, EXIT
5621 031114 012737 000016 002334 IOV #IENR<17>,REGNUM ;NO! BUILD REGISTER # POINTER

```


5672

.SBTTL TEST 27 VIA TIMER # 1 ONE SHOT MODE

```

.....
:
: TEST 27 VIA TIMER # 1 ONE SHOT MODE
:
: THIS TEST VERIFIES THAT THE TIMER 1 COUNTER IS OPERATIONAL IN ONE-SHOT
: MODE, IN EACH OF TWO SUBTESTS.
:
: IN THE FIRST SUBTEST, THE FOLLOWING IS PERFORMED :
:
: A MASTER CLEAR IS DONE AND THE TIMER IS PLACED IN ONE-SHOT MODE BY
: SETTING BOTH ACR7 & ACR6 TO 0.
:
: THE PROGRAM CHECKS FOR THE "T1" FLAG (BIT 6) IN THE IFR TO BE
: INITIALLY CLEARED.
:
: T1L-L (ADR 04) & T1C-H (ADR 05) ARE BOTH LOADED WITH 252 (OCTAL).
: (THIS IS EQUIVALENT TO AAAA (HEX) OR 43,690 (DECIMAL).) LOADING
: T1C-H STARTS THE COUNTER.
:
: THE PROGRAM PERIODICALLY CHECKS THE COUNTER TO VERIFY THAT IT IS
: DECREMENTING AND THAT IT EVENTUALLY UNDERFLOWS PAST 0 AND CONTINUES
: TO DECREMENT.
:
: T1L L (ADR 04) IS LOADED WITH 001 & T1C-H (ADR 05) IS LOADED WITH
: 000 IN ORDER TO SET "T1" WITH A QUICK UNDERFLOW. THE 'T1' FLAG BIT
: IN IFR IS READ AND CHECKED TO BE SET.
:
: T1C H, T1L-L, & T1L-H (ADDR'S 05, 06, & 07 RESP.) ARE READ AND AFTER
: EACH THE "T1" INTERRUPT FLAG IS CHECKED TO BE STILL SET.
:
: T1C L (ADDR 04) IS READ AND "T1" IS CHECKED TO BE CLEARED.
:
: T1C-H IS LOADED WITH 0 AGAIN TO INITIATE A NEW COUNT DOWN (WHICH
: SHOULD UNDERFLOW ALMOST IMMEDIATELY) AND THE 'T1' BIT IN IFR IS
: CHECKED TO BE SET AGAIN.
:
: T1L-L IS LOADED WITH 125 (OCTAL) AND "T1" BIT IS CHECKED TO BE STILL
: SET.
:
: T1C H IS LOADED WITH 125, AND THE "T1" BIT IS READ AND CHECKED TO BE
: CLEARED BY THE LOADING OF T1C-H.
:
: IN THE SECOND SUBTEST, ALL OF THE ABOVE OPERATIONS ARE REPEATED,
: WITH ACR BIT 7 SET TO 1, AND ACR BIT 6 SET TO 0. "PB7" (BIT 7 OF ORB)
: WILL BE MONITORED FOR ITS EXPECTED LEVELS DURING THIS SUBTEST.
:

```

```

.....
:
: BGN1ST
:
: BGNSUB
:
: T27::
:
: T27.1: TRAP C1BSUB
:

```

5673 031134
031134
031134
031134 104402
5674

TEST 27 VIA TIMER # 1 ONE SHOT MODE

5675									
5676	031136	004737	003762		JSR	PC,MSTCLR		;INIT DMV & ENTER M LOOP	
5677	031142	103003			BCC	18		;IF NO ERROR, PROCEED WITH TESTING	
5678	031144				ERROR			;ELSE, REPORT ERROR	
	031144	104460							TRAP C\$ERROR
5679	031146				ESCAPE	TST		; & EXIT TEST	
	031146	104410							TRAP C\$ESCAPE
	031150	004742							.WORD L10073 .
5680	031152	004537	004660	18:	JSR	R5,INITT1		;INITIALIZE TIMER # 1	
5681	031156	000000			0			; 0 ==> LATCHES	
5682	031160	000000			0			; MODE 0 & "T1" INT. ENABLE FLAG CLEARED	
5683	031162	103003			BCC	.+10		;IF NO ERROR, PROCEED	
5684	031164				ERROR			;ELSE, REPORT IT	
	031164	104460							TRAP C\$ERROR
5685	031166				ESCAPE	TST		; AND EXIT THIS TEST	
	031166	104410							TRAP C\$ESCAPE
	031170	004722							.WORD L10073-.
5686	031172	004737	036146		JSR	PC,GETT1		;IS "T1" SET?	
5687	031176	102002			BVC	.+6		;IF NO ERROR, PROCEED	
5688	031200				ESCAPE	SUB		;ELSE, IT'S ALREADY BEEN REPORTED	EXIT
	031200	104410							TRAP C\$ESCAPE
	031202	002202							.WORD L10074-.
5689	031204	103143			BCC	68		;NO, GOOD.	
5690	031206	004537	004064		JSR	R5,READ		;GET T1CL FOR ERROR MESSAGE	
5691	031212	120004			T1CL				
5692	031214	002450			TMP4				
5693	031216	103003			BCC	.+10		;IF NO ERROR, PROCEED	
5694	031220				ERROR			;ELSE, REPORT IT	
	031220	104460							TRAP C\$ERROR
5695	031222				ESCAPE	TST		; AND EXIT THIS TEST	
	031222	104410							TRAP C\$ESCAPE
	031224	004666							.WORD L10073-.
5696	031226	004537	004064		JSR	R5,READ		;GET T1CH FOR ERROR MESSAGE	
5697	031232	120005			T1CH				
5698	031234	002452			TMP5				
5699	031236	103003			BCC	.+10		;IF NO ERROR, PROCEED	
5700	031240				ERROR			;ELSE, REPORT IT	
	031240	104460							TRAP C\$ERROR
5701	031242				ESCAPE	TST		; AND EXIT THIS TEST	
	031242	104410							TRAP C\$ESCAPE
	031244	004646							.WORD L10073-.
5702	031246	004537	004064		JSR	R5,READ		;GET T1LL FOR ERROR MESSAGE	
5703	031252	120006			T1LL				
5704	031254	002454			TMP6				
5705	031256	103003			BCC	.+10		;IF NO ERROR, PROCEED	
5706	031260				ERROR			;ELSE, REPORT IT	
	031260	104460							TRAP C\$ERROR
5707	031262				ESCAPE	TST		; AND EXIT THIS TEST	
	031262	104410							TRAP C\$ESCAPE
	031264	004626							.WORD L10073 .
5708	031266	004537	004064		JSR	R5,READ		;GET T1LH FOR ERROR MESSAGE	
5709	031272	120007			T1LH				
5710	031274	002456			TMP7				
5711	031276	103003			BCC	.+10		;IF NO ERROR, PROCEED	
5712	031300				ERROR			;ELSE, REPORT IT	
	031300	104460							TRAP C\$ERROR
5713	031302				ESCAPE	TST		; AND EXIT THIS TEST	

TEST 27 VIA TIMER # 1 ONE SHOT MODE

```

031302 104410
031304 004606
5714 031306 004537 004064 JSR R5,READ ;GET ACR FOR ERROR MESSAGE TRAP C$ESCAPE
031312 120013 ACR ;.WORD L10073 .
5715 031312 120013 TMPB
5716 031314 002466 BCC .+10 ;IF NO ERROR, PROCEED
5717 031316 103003 ERROR ;ELSE, REPORT IT
5718 031320 104460
031322 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
031324 004566 ;.WORD C$ESCAPE
5720 031326 GEDF EM50A,ERR50 ;YES, REPORT IT'S NOT BEING CLEARED @ INIT. ; 'DEVICE FATAL' ERROR # 50 ;.WORD L10073 .
031326 104455 TRAP C$ERDF
031330 000062 ;.WORD 50
031332 016067 ;.WORD EM50A
031334 010762 ;.WORD ERR50
5721 031336 PRINTX #FMT50M ; & SAY THE COUNTERS HAVEN'T BEEN LOADED YET! ;.WORD
031336 012746 013174 MOV #FMT50M, -(SP)
031342 012746 000001 MOV #1, (SP)
031346 010600 MOV SP,R0
031350 104415 TRAP C$PNTX
031352 062706 000004 ADD #4,SP
5722
5723
5724
5725 031356 112737 000002 002453 MOVB #2,TMP5+1
5726 031364 004537 004310 JSR R5,WRITE ;INIT TIMER # 1 BY WRITING INTO
5727 031370 120005 T1CH ;T1C H (ADDR 05)
5728 031372 002453 TMP5+1
5729 031374 103003 BCC .+10 ;IF NO ERROR, PROCEED
5730 031376 ERROR ;ELSE, REPORT IT
5731 031376 104460
031400 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
031402 004510 ;.WORD C$ESCAPE
5732 031404 004737 036146 JSR PC,GETT1 ;IS "T1" SET? TRAP C$ESCAPE
5733 031410 102002 BVC .+6 ;IF NO ERROR, PROCEED ;.WORD L10073 .
5734 031412 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
031412 104410 TRAP C$ESCAPE
031414 001770 ;.WORD L10074 .
5735 031416 103036
5736 031420 004537 004064 BCC 6$ ;NO, GOOD.
5737 031424 120004 JSR R5,READ ;GET T1CL FOR ERROR MESSAGE
5738 031426 002450 T1CL
5739 031430 103003 TMP4
5740 031432 BCC .+10 ;IF NO ERROR, PROCEED
031432 104460 ERROR ;ELSE, REPORT IT
5741 031434 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
031434 104410 ;.WORD C$ESCAPE
031436 004454 ;.WORD L10073 .
5742 031440 004537 004064 JSR R5,READ ;GET T1CH FOR ERROR MESSAGE
5743 031444 120005 T1CH
5744 031446 002452 TMP5
5745 031450 103003 BCC .+10 ;IF NO ERROR, PROCEED
5746 031452 104460 ERROR ;ELSE, REPORT IT
; TRAP C$ERROR

```


TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

SEQ 0160

```

5747 031454          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ESCAPE
      031454 104410          ;          ;          .WORD      L10073-.
      031456 004434          ;          ;          ;          ;
5748 031460 004537 004064 JSR      R5,READ          ;GET T1LM FOR ERROR MESSAGE
5749 031464 120007          T1LM
5750 031466 002456          TMP7
5751 031470 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5752 031472          ERROR          ;ELSE, REPORT IT
      031472 104460          ;          ;          TRAP      C$ERROR
5753 031474          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ESCAPE
      031474 104410          ;          ;          .WORD      L10073 .
      031476 004414          ;          ;          ;          ;
5754 031500          GEDF      EM508,ERR50      ;YES, REPORT IT'S NOT BEING CLEARED @ INIT.
      ;          ;          ;          "DEVICE FATAL" ERROR # 51
      ;          ;          ;          ;          TRAP      C$ERDF
      ;          ;          ;          ;          .WORD      51
      ;          ;          ;          ;          .WORD      EM508
      ;          ;          ;          ;          .WORD      ERR50
      031500 104455          ;          ;          ;          ;
      031502 000063          ;          ;          ;          ;
      031504 016135          ;          ;          ;          ;
      031506 010762          ;          ;          ;          ;
5755 031510          ESCAPE SUB          ;AND EXIT SUBTEST          TRAP      C$ESCAPE
      031510 104410          ;          ;          .WORD      L10074 .
      031512 001672          ;          ;          ;          ;
5756
5757          ;          ;          ;          ;
5758          ;          ;          ;          ;
5759 031514 004537 004064 6$: JSR      R5,READ          ;GET ACR FOR LATER ERROR MESSAGES
5760 031520 120013          ACR
5761 031522 002466          TMPB
5762 031524 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5763 031526          ERROR          ;ELSE, REPORT IT
      031526 104460          ;          ;          TRAP      C$ERROR
5764 031530          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ESCAPE
      031530 104410          ;          ;          .WORD      L10073 .
      031532 004360          ;          ;          ;          ;
5765 031534 112737 000377 002445 MOVB     #377,TMP2+1      ;INITIALIZE ORB FOR INPUT/OUTPUT
5766 031542 004537 004310 JSR      R5,WRITE
5767 031546 120002          DDRB
5768 031550 002445          TMP2+1
5769 031552 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5770 031554          ERROR          ;ELSE, REPORT IT
      031554 104460          ;          ;          TRAP      C$ERROR
5771 031556          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ESCAPE
      031556 104410          ;          ;          .WORD      L10073 .
      031560 004332          ;          ;          ;          ;
5772 031562 112737 000377 002441 MOVB     #377,TMP0+1      ;SETUP VALUE FOR ORB
5773 031570 004537 004310 JSR      R5,WRITE          ;DO IT
5774 031574 120000          ORB
5775 031576 002441          TMP0+1
5776 031600 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5777 031602          ERROR          ;ELSE, REPORT IT
      031602 104460          ;          ;          TRAP      C$ERROR
5778 031604          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ESCAPE
      031604 104410          ;          ;          .WORD      L10073 .
      031606 004304          ;          ;          ;          ;
5779 031610 004537 036114 JSR      R5,LODT1C          ;LOAD TIMER # 1
5780 031614          .BYTE      252
5781 031615          .BYTE      252
5782

```

```

5783
5784
5785 031616 004737 036332      JSR    PC,GETPB7      ;GET "PB7". IS IT SET?
5786 031622 102002              BVC    .+6            ;IF NO ERROR, PROCEED
5787 031624              ESCAPE  SUB            ;ELSE, IT'S ALREADY BEEN REPORTED      EXIT
                                     TRAP    C$ESCAPE
                                     .WORD  L10074-.
5788 031630 103404              BCS    9$            ;IT IS. (SET BY LOADING DDRB & ORB ABOVE)
5789 031632              GEDF   EMSOW,ERR1    ;IF NOT, TIMER 1 CLEARED IT!!!
                                     ; "DEVICE FATAL" ERROR # 52
                                     TRAP    C$ERDF
                                     .WORD  52
                                     .WORD  EMSOW
                                     .WORD  ERR1
          031632 104455
          031634 000064
          031636 017305
          031640 005276
5790 031642 004737 036204      9$:   JSR    PC,KICKT1 ;IT TAKES SO MUCH TIME TO CHECK FOR (E MAYBE
5791                                     ; EVEN REPORT) THIS ERROR THAT WE SHOULD
5792                                     ; KICK OFF THE TIMER AGAIN
5793 031646 103003              BCC    .+10          ;IF NO ERROR, PROCEED
5794 031650              ERROR  .+10          ;ELSE, REPORT IT
5795 031652              ESCAPE  TST          ; AND EXIT THIS TEST      TRAP    C$ERROR
          031652 104410
          031654 004236
          031656 000240              NOP
5796 031656 000240              NOP
5797 031660 000240              NOP
5798 031662 000240              JSR    R5,READ      ;NOW WAIT A LITTLE WHILE - LET T1CL CHANGE
5799 031664 004537 004064      T1CL
5800 031670 120004              TMP4
5801 031672 002450              BCC    .+10          ;IF NO ERROR, PROCEED
5802 031674 103003              ERROR  .+10          ;ELSE, REPORT IT
5803 031676              ESCAPE  TST          ; AND EXIT THIS TEST      TRAP    C$ERROR
          031676 104460
          031700 104410
          031702 004210
5805 031704 123737 002450 031614  CMPB   TMP4,7$      ;MAKE SURE THE COUNTER IS DECREMENTING
5806 031712 001024              BNE    12$            ;IT IS, NOW SEE IF THE HIGH COUNTER IS TOO
5807 031714 004537 004064      JSR    R5,READ      ;GET T1CH FOR ERROR MESSAGE
5808 031720 120005              T1CH
5809 031722 002452              TMP5
5810 031724 103003              BCC    .+10          ;IF NO ERROR, PROCEED
5811 031726              ERROR  .+10          ;ELSE, REPORT IT
          031726 104460
5812 031730              ESCAPE  TST          ; AND EXIT THIS TEST      TRAP    C$ERROR
          031730 104410
          031732 004160
5813 031734 004537 004064      JSR    R5,READ      ;GET T1LL FOR ERROR MESSAGE
          031734 004160
          031740 120006              T1LL
5814 031740 120006              TMP6
5815 031742 002454              BCC    .+10          ;IF NO ERROR, PROCEED
5816 031744 103003              ERROR  .+10          ;ELSE, REPORT IT
          031744 104460
5817 031746              ESCAPE  TST          ; AND EXIT THIS TEST      TRAP    C$ERROR
          031746 104410
          031750 104410
          031752 004140
5818 031750              ESCAPE  TST          ; AND EXIT THIS TEST      TRAP    C$ERROR
          031750 104410
          031752 004140
5819 031754              GEDF   EMS0D,ERR50 ;IT WASN'T - REPORT THE ERROR
                                     ; "DEVICE FATAL" ERROR # 53
                                     TRAP    C$ESCAPE
                                     .WORD  L10073.

```


TEST 27 VIA TIMER # 1 ONE SHOT MODE

```

5854 032130 120002          DDRB
5855 032132 002445          TMP2+1
5856 032134 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5857 032136          ERROR          ;ELSE, REPORT IT
5858 032140          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
          032140 104410          ;          TRAP      C$ESCAPE
          032142 003750          .WORD      L10073 .
5859 032144 023737 002440 002441  CMP      TMP0,TMP0+1          ;CLEAR PB7 BY WRITING INTO ORB
5860 032152 004537 004322          JSR      R5,WRITEI
5861 032156 120000          ORB
5862 032160 000030          30          ; (THIS CLEARS DTR & RTS! ALSO)
5863 032162 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5864 032164          ERROR          ;ELSE, REPORT IT
5865 032166          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
          032166 104410          ;          TRAP      C$ESCAPE
          032170 003722          .WORD      L10073 .
5866 032172 004537 036114          JSR      R5,LODT1C          ;RE LOAD TIMER # 1 WITH A VALUE WHICH CAUSE AN
5867 032176          001          18$: .BYTE 1          ;ALMOST IMMEDIATE TIMEOUT
5868 032177          000          19$: .BYTE 0          ; (ADDRESS OF HIGH BYTE FOR TIC-H (ADDR 05))
5869          ;-----
5870
5871 032200 004737 036146          JSR      PC,GETT1          ;WAS "T1" SET BY THE ABOVE OPERATION?
5872 032204 102002          BVC      .+6          ;IF NO ERROR, PROCEED
5873 032206          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
          032206 104410          ;          TRAP      C$ESCAPE
          032210 001174          .WORD      L10074 .
5874 032212 103426          BCS      20$          ;YES, OK -- CONTINUE ERROR CHECKING
5875 032214 004537 004064          JSR      R5,READ          ;GET TILL FOR ERROR MESSAGE
5876 032220 120006          TILL
5877 032222 002454          TMP6
5878 032224 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5879 032226          ERROR          ;ELSE, REPORT IT
5880 032230          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
          032230 104410          ;          TRAP      C$ESCAPE
          032232 003660          .WORD      L10073 .
5881 032234 004537 004064          JSR      R5,READ          ;GET T1LH FOR ERROR MESSAGE
5882 032240 120007          T1LH
5883 032242 002456          TMP7
5884 032244 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5885 032246          ERROR          ;ELSE, REPORT IT
5886 032246 104460          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
          032250 104410          ;          TRAP      C$ESCAPE
          032252 003640          .WORD      L10073 .
5887 032254          GEDF      EM50F,ERR50          ;NO, BAD NEWS! REPORT THE FAILURE
          ;          "DEVICE FATAL" ERROR # 55
          ;          TRAP      C$ERDF
          ;          .WORD      55
          ;          .WORD      EM50F
          ;          .WORD      ERR50
5888 032264          ESCAPE SUB          ;          AND GET OUT OF SUBTEST          TRAP      C$ESCAPE
          032264 104410          ;          TRAP      C$ESCAPE
          032266 001116          .WORD      L10074 .
5889 032270 004737 036332          20$: JSR      PC,GETFB7          ;GET 'PB7'. IS IT CLEARED?

```

TEST 27 -- VIA TIMER # 1 ONE-SHOT MODE

```

5890 032274 102002          BVC      .+6      ;IF NO ERROR, PROCEED
5891 032276          ESCAPE  SUB      ;ELSE, IT'S ALREADY BEEN REPORTED EXIT
      032276 104410          TRAP      C$ESCAPE
      032300 001104          .WORD    L10074 .
5892 032302 103024          BCC      40$      ;IF CLEARED, DDRB IS STILL IN CONTROL OF IT
5893 032304 004537 004064  JSR      R5,READ  ;GET TILL FOR ERROR MESSAGE
5894 032310 120006          TILL
5895 032312 002454          TMP6
5896 032314 103003          BCC      .+10     ;IF NO ERROR, PROCEED
5897 032316          ERROR      ;ELSE, REPORT IT
      032316 104460          TRAP      C$ERROR
5898 032320          ESCAPE  TST      ;      AND EXIT THIS TEST
      032320 104410          TRAP      C$ESCAPE
      032322 003570          .WORD    L10073 .
5899 032324 004537 004064  JSR      R5,READ  ;GET TILH FOR ERROR MESSAGE
5900 032330 120007          TILH
5901 032332 002456          TMP7
5902 032334 103003          BCC      .+10     ;IF NO ERROR, PROCEED
5903 032336          ERROR      ;ELSE, REPORT IT
      032336 104460          TRAP      C$ERROR
5904 032340          ESCAPE  TST      ;      AND EXIT THIS TEST
      032340 104410          TRAP      C$ESCAPE
      032342 003550          .WORD    L10073-.
5905 032344          GEDF    EMSOW,ERR50 ;ELSE, IT'S BEING SET BY TIMER 1 IN MODE 0!
      032344 104455          ;      "DEVICE FATAL" ERROR # 56
      032346 000070          TRAP      C$ERDF
      032350 017305          .WORD    56
      032352 010762          .WORD    EMSOW
      032352 010762          .WORD    ERR50
5906 032354 004537 004064  40$: JSR      R5,READ  ;READ TIC-H (ADDR 05) TO SEE IF THIS CLEARS "T1"
5907 032360 120005          T1CH      ;(THIS VALUE ISN'T CHECKED BECAUSE IT CAN BE
5908 032362 002452          TMP5      ;ALMOST ANYTHING)
5909 032364 103003          BCC      .+10     ;IF NO ERROR, PROCEED
5910 032366          ERROR      ;ELSE, REPORT IT
      032366 104460          TRAP      C$ERROR
5911 032370          ESCAPE  TST      ;      AND EXIT THIS TEST
      032370 104410          TRAP      C$ESCAPE
      032372 003520          .WORD    L10073-.
5912 032374 004737 036146  JSR      PC,GETT1 ;PUT THE CURRENT "T1" VALUE INTO THE CARRY BIT
5913 032400 102002          BVC      .+6      ;IF NO ERROR, PROCEED
5914 032402          ESCAPE  SUB      ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
      032402 104410          TRAP      C$ESCAPE
      032404 001000          .WORD    L10074 .
5915 032406 103425          BCS      21$      ;IF SET, READING T1CH DIDN'T CLEAR IT - OK!
5916 032410 004537 004064  JSR      R5,READ  ;GET TILL FOR ERROR MESSAGE
5917 032414 120006          TILL
5918 032416 002454          TMP6
5919 032420 103003          BCC      .+10     ;IF NO ERROR, PROCEED
5920 032422          ERROR      ;ELSE, REPORT IT
      032422 104460          TRAP      C$ERROR
5921 032424          ESCAPE  TST      ;      AND EXIT THIS TEST
      032424 104410          TRAP      C$ESCAPE
      032426 003464          .WORD    L10073 .
5922 032430 004537 004064  JSR      R5,READ  ;GET TILH FOR ERROR MESSAGE
5923 032434 120007          TILH
5924 032436 002456          TMP7
5925 032440 103003          BCC      .+10     ;IF NO ERROR, PROCEED

```

TEST 27 - VIA TIMER # 1 ONE SHOT MODE

```

5926 032442          ERROR          ;ELSE, REPORT IT
      032442 104460          TRAP      C$ERROR
5927 032444          ESCAPE TST      ;      AND EXIT THIS TEST      TRAP      C$ESCAPE
      032444 104410          ;                                      .WORD      L10073 .
      032446 003444          ;
5928 032450          GEDF      EM50G,ERR50 ;IF CLEARED! BAD VIA CHIP!
      032450 104455          ;      "DEVICE FATAL" ERROR # 57
      032452 000071          ;                                      TRAP      C$ERDF
      032454 016406          ;                                      .WORD      57
      032456 010762          ;                                      .WORD      EM50G
      032460 000507          ;                                      .WORD      ERR50
5929 032460          BR      28$      ;BYPASS THE REST OF THIS SECTION OF TESTING
5930
5931 032462 004537 004064 21$: JSR      R5,READ      ;READ T1L-L (ADDR 06)
5932 032466 120006          T1LL
5933 032470 002454          TMP6
5934 032472 103003          BCC      .+10      ;THIS SHOULD RETURN A 001
5935 032474          ERROR          ;IF NO ERROR, PROCEED
      032474 104460          ;ELSE, REPORT IT
5936 032476          ESCAPE TST      ;      AND EXIT THIS TEST      TRAP      C$ERROR
      032476 104410          ;                                      TRAP      C$ESCAPE
      032500 003412          ;                                      .WORD      L10073 .
5937 032502 123737 002454 032176 CMPB     TMP6,18$ ;CHECK T1L L (ADDR 06) AGAINST LOADED VALUE
5938 032510 001415          BEQ      23$
5939 032512 004537 004064 JSR      R5,READ      ;IF SAME, PROCEED
5940 032516 120007          T1LH
5941 032520 002456          TMP7
5942 032522 103003          BCC      .+10      ;GET T1LH FOR ERROR MESSAGE
5943 032524          ERROR          ;IF NO ERROR, PROCEED
      032524 104460          ;ELSE, REPORT IT
5944 032526          ESCAPE TST      ;      AND EXIT THIS TEST      TRAP      C$ERROR
      032526 104410          ;                                      TRAP      C$ESCAPE
      032530 003362          ;                                      .WORD      L10073-.
5945 032532          GEDF      EM50H,ERR50 ;ELSE, REPORT BAD LOAD OF T1L-L (ADDR 06)
      032532 104455          ;      "DEVICE FATAL" ERROR # 58
      032534 000072          ;                                      TRAP      C$ERDF
      032536 016450          ;                                      .WORD      58
      032540 010762          ;                                      .WORD      EM50H
      032542 000456          ;                                      .WORD      ERR50
5946 032542          BR      28$      ;BYPASS THE REST OF THIS SECTION OF TESTING
5947
5948 032544 004737 036146 23$: JSR      PC,GETT1 ;IS "T1" STILL SET?
5949 032550 102002          BVC
5950 032552          ESCAPE SUB      ;IF NO ERROR, PROCEED
      032552 104410          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
      032554 000630          ;                                      TRAP      C$ESCAPE
      032556 103415          ;                                      .WORD      L10074 .
5951 032556 004537 004064 BCS      24$      ;YES, ALL'S OK
5952 032560 120007          JSR      R5,READ      ;GET T1LH FOR ERROR MESSAGE
5953 032564 002456          T1LH
5954 032566 103003          TMP7
5955 032570 104460          BCC      .+10      ;IF NO ERROR, PROCEED
5956 032572          ERROR          ;ELSE, REPORT IT
      032572 104460          ;
5957 032574          ESCAPE TST      ;      AND EXIT THIS TEST      TRAP      C$ERROR
      032574 104410          ;                                      TRAP      C$ESCAPE
      032576 003314          ;                                      .WORD      L10073 .
5958 032600          GEDF      EM50I,ERR50 ;NO! BAD VIA CHIP!

```

TEST 27 -- VIA TIMER # 1 ONE SHOT MODE

```

032600 104455
032602 000073
032604 016536
032606 010762
5959 032610 000433 BR 28$ ;"DEVICE FATAL" ERROR # 59
5960 ; TRAP C$ERDF
; .WORD 59
; .WORD EM50I
; .WORD ERR50
;BYPASS THE REST OF THIS SECTION OF TESTING
5961 032612 004537 004064 24$: JSR R5,READ ;READ T1L-H (ADDR 07)
5962 032616 120007 T1LH
5963 032620 002456 TMP7 ;THIS SHOULD RETURN A 000
5964 032622 103003 BCC .+10 ;IF NO ERROR, PROCEED
5965 032624 ERROR ;ELSE, REPORT IT
032624 104460 TRAP C$ERROR
5966 032626 ESCAPE TST ; AND EXIT THIS TEST
; .WORD L10073
032626 104410 TRAP C$ESCAPE
032630 003262 .WORD L10073
5967 032632 123737 002456 032177 CMPB TMP7,19$ ;CHECK T1L-H (ADDR 07) AGAINST LOADED VALUE
5968 032640 001405 BEQ 26$ ;IF SAME, PROCEED
5969 032642 GEDF EM50J,ERR50 ;ELSE, REPORT BAD LOAD OF T1L H (ADDR 07)
; "DEVICE FATAL" ERROR # 60
032642 104455 TRAP C$ERDF
032644 000074 .WORD 60
032646 016600 .WORD EM50J
032650 010762 .WORD ERR50
5970 032652 000412 BR 28$ ;BYPASS THE REST OF THIS SECTION OF TESTING
5971
5972 032654 004737 036146 26$: JSR PC,GETT1 ;IS "T1" STILL SET?
5973 032660 102002 BVC .+6 ;IF NO ERROR, PROCEED
5974 032662 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
032662 104410 TRAP C$ESCAPE
032664 000520 .WORD L10074-.
5975 032666 103404 BCS 28$ ;YES, ALL'S OK
5976 032670 GEDF EM50K,ERR50 ;NO! BAD VIA CHIP!
; "DEVICE FATAL" ERROR # 61
032670 104455 TRAP C$ERDF
032671 000075 .WORD 61
032674 016666 .WORD EM50K
032676 010762 .WORD ERR50
5977
5978
5979 ;-----
5980 032700 004537 004064 28$: JSR R5,READ ;READ T1C-L (ADDR 04)
5981 032704 120004 T1CL ;(THIS VALUE ISN'T CHECKED BECAUSE IT CAN BE
5982 032706 002450 TMP4 ;ALMOST ANYTHING)
5983 032710 103003 BCC .+10 ;IF NO ERROR, PROCEED
5984 032712 ERROR ;ELSE, REPORT IT
032712 104460 TRAP C$ERROR
5985 032714 ESCAPE TST ; AND EXIT THIS TEST
; .WORD L10073
032714 104410 TRAP C$ESCAPE
032716 003174 .WORD L10073
5986 032720 004737 036146 JSR PC,GETT1 ;IS "T1" CLEARED NOW
5987 032724 102002 BVC .+6 ;IF NO ERROR, PROCEED
5988 032726 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
032726 104410 TRAP C$ESCAPE
032730 000454 .WORD L10074-.
5989 032732 103024 BCC 29$ ;YES, ALL'S OK
5990 032734 004537 004064 JSR R5,READ ;GET T1L FOR ERROR MESSAGE

```

TEST 27 VIA TIMER # 1 ONE-SHOT MODE

```

5991 032740 120006          TILL
5992 032742 002454          TMP6
5993 032744 103003          BCC      .+10          ;IF NO ERROR, PROCEED
5994 032746          ERROR          ;ELSE, REPORT IT
          032746 104460          TRAP      C$ERROR
5995 032750          ESCAPE TST          ;          AND EXIT THIS TEST
          032750 104410          TRAP      C$ESCAPE
          032752 003140          .WORD    L10073 .
5996 032754 004537 004064    JSR      RS,READ        ;GET T1LH FOR ERROR MESSAGE
5997 032760 120007          T1LH
5998 032762 002456          TMP7
5999 032764 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6000 032766          ERROR          ;ELSE, REPORT IT
          032766 104460          TRAP      C$ERROR
6001 032770          ESCAPE TST          ;          AND EXIT THIS TEST
          032770 104410          TRAP      C$ESCAPE
          032772 003120          .WORD    L10073-.
6002 032774          GEDF   EM50C,ERR50 ;NO!  BAD VIA CHIP!
          ;          "DEVICE FATAL" ERROR # 62
          032774 104455          TRAP      C$ERDF
          032776 000076          .WORD    62
          033000 016203          .WORD    EM50C
          033002 010762          .WORD    ERR50
6003
6004
6005          ;-----
6006 033004 004537 004310    29$:  JSR      RS,WRITE    ;RE-WRITE INTO T1C H (ADDR 05) TO SET T1 AGAIN
6007 033010 120005          T1CH
6008 033012 002453          TMP5+1
6009 033014 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6010 033016          ERROR          ;ELSE, REPORT IT
          033016 104460          TRAP      C$ERROR
6011 033020          ESCAPE TST          ;          AND EXIT THIS TEST
          033020 104410          TRAP      C$ESCAPE
          033022 003070          .WORD    L10073 .
6012 033024 004737 036146    JSR      PC,GETT1      ;IS "T1" SET AGAIN
6013 033030 102002          BVC     .+6          ;IF NO ERROR, PROCEED
6014 033032          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
          033032 104410          TRAP      C$ESCAPE
          033034 000350          .WORD    L10074 .
6015 033036 103426          BCS     32$          ;YES, ALL'S WELL (AGAIN?)
6016 033040 004537 004064    JSR      RS,READ        ;GET T1LH FOR ERROR MESSAGE
6017 033044 120006          T1LH
6018 033046 002454          TMP6
6019 033050 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6020 033052          ERROR          ;ELSE, REPORT IT
          033052 104460          TRAP      C$ERROR
6021 033054          ESCAPE TST          ;          AND EXIT THIS TEST
          033054 104410          TRAP      C$ESCAPE
          033056 003034          .WORD    L10073-.
6022 033060 004537 004064    JSR      RS,READ        ;GET T1LH FOR ERROR MESSAGE
6023 033064 120007          T1LH
6024 033066 002456          TMP7
6025 033070 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6026 033072          ERROR          ;ELSE, REPORT IT
          033072 104460          TRAP      C$ERROR
6027 033074          ESCAPE TST          ;          AND EXIT THIS TEST

```


TEST 27 VIA TIMER # 1 ONE SHOT MODE

```

033074 104410 TRAP C$ESCAPE
033076 003014 .WORD L10073-.
6028 033100 GEDF EMSOL,ERR50 ;NO! SOMETHING WENT WRONG! REPORT IT
; "DEVICE FATAL" ERROR # 63
033100 104455 TRAP C$ERDF
033102 000077 .WORD 63
033104 016730 .WORD EMSOL
033106 010762 .WORD ERR50
6029 033110 ESCAPE SUB ; AND EXIT FROM THIS SUBTEST
033110 104410 TRAP C$ESCAPE
033112 000272 .WORD L10074-.

6030
6031 ;-----
6032
6033 033114 1127, ' 000125 002455 32$: MOVB #125,TMP6+1 ;USING A DIFFERENT VALUE - 55 HEX.,
6034 033122 004537 004310 JSR R5,WRITE ;RE LOAD TIL-L (ADDR 06)
6035 033126 120006 TILL
6036 033130 002455 TMP6+1
6037 033132 103003 BCC .+10 ;IF NO ERROR, PROCEED
6038 033134 ERROR ;ELSE, REPORT IT
033134 104460 TRAP C$ERROR
6039 033136 ESCAPE TST ; AND EXIT THIS TEST
033136 104410 TRAP C$ESCAPE
033140 002752 .WORD L10073-.
6040 033142 004737 036146 JSR PC,GETT1 ;IS "T1" STILL SET?
6041 033146 102002 BVC .+6 ;IF NO ERROR, PROCEED
6042 033150 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
033150 104410 TRAP C$ESCAPE
033152 000232 .WORD L10074 .
6043 033154 103426 BCS 33$ ;YES, ALL'S STILL OK
6044 033156 004537 004064 JSR R5,READ ;GET TILL FOR ERROR MESSAGE
6045 033162 120006 TILL
6046 033164 002454 TMP6
6047 033166 103003 BCC .+10 ;IF NO ERROR, PROCEED
6048 033170 ERROR ;ELSE, REPORT IT
033170 104460 TRAP C$ERROR
6049 033172 ESCAPE TST ; AND EXIT THIS TEST
033172 104410 TRAP C$ESCAPE
033174 002716 .WORD L10073-.
6050 033176 004537 004064 JSR R5,READ ;GET TILH FOR ERROR MESSAGE
6051 033202 120007 TILH
6052 033204 002456 TMP7
6053 033206 103003 BCC .+10 ;IF NO ERROR, PROCEED
6054 033210 ERROR ;ELSE, REPORT IT
033210 104460 TRAP C$ERROR
6055 033212 ESCAPE TST ; AND EXIT THIS TEST
033212 104410 TRAP C$ESCAPE
033214 002676 .WORD L10073 .
6056 033216 GEDF EMSOM,ERR50 ;NO! SOMETHING WENT WRONG! REPORT IT
; "DEVICE FATAL" ERROR # 64
033216 104455 TRAP C$ERDF
033220 000100 .WORD 64
033222 017012 .WORD EMSOM
033224 010762 .WORD ERR50
6057 033226 ESCAPE SUB ; AND EXIT FROM THIS SUBTEST
033226 104410 TRAP C$ESCAPE
033230 000154 .WORD L10074 .

```

```

6058
6059
6060
6061 033232 112737 000125 002453 33$: MOVB #125,TMP5+1 ;AND USING THE SAME VALUE AGAIN (55 HEX),
6062 033240 004537 004310 JSR R5,WRITE ;NOW LOAD T1C-H (ADDR 05)
6063 033244 120005 T1CH
6064 033246 002453 TMP5+1
6065 033250 103003 BCC .+10 ;IF NO ERROR, PROCEED
6066 033252 104460 ERROR ;ELSE, REPORT IT
6067 033254 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033254 104410 ; TRAP C$ESCAPE
033256 002634 .WORD L10073 .
6068 033260 004737 036146 JSR PC,GETT1 ;"T1" SHOULD NOW BE CLEARED
6069 033264 102002 BVC .+6 ;IF NO ERROR, PROCEED
6070 033266 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
033266 104410 TRAP C$ESCAPE
033270 000114 .WORD L10074-.
6071 033272 103044 BCC 34$ ;IT WAS, ALL'S WELL THAT END'S WELL (I THINK!?)
6072 033274 004537 004064 JSR R5,READ ;GET T1CL FOR ERROR MESSAGE
6073 033300 120004 T1CL
6074 033302 002450 TMP4
6075 033304 103003 BCC .+10 ;IF NO ERROR, PROCEED
6076 033306 104460 ERROR ;ELSE, REPORT IT
6077 033310 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033310 104410 ; TRAP C$ESCAPE
033312 002600 .WORD L10073-.
6078 033314 004537 004064 JSR R5,READ ;GET T1CH FOR ERROR MESSAGE
6079 033320 120005 T1CH
6080 033322 002452 TMP5
6081 033324 103003 BCC .+10 ;IF NO ERROR, PROCEED
6082 033326 104460 ERROR ;ELSE, REPORT IT
6083 033330 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033330 104410 ; TRAP C$ESCAPE
033332 002560 .WORD L10073-.
6084 033334 004537 004064 JSR R5,READ ;GET T1LL FOR ERROR MESSAGE
6085 033340 120006 T1LL
6086 033342 002454 TMP6
6087 033344 103003 BCC .+10 ;IF NO ERROR, PROCEED
6088 033346 104460 ERROR ;ELSE, REPORT IT
6089 033350 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033350 104410 ; TRAP C$ESCAPE
033352 002540 .WORD L10073 .
6090 033354 004537 004064 JSR R5,READ ;GET T1LH FOR ERROR MESSAGE
6091 033360 120007 T1LH
6092 033362 002456 TMP7
6093 033364 103003 BCC .+10 ;IF NO ERROR, PROCEED
6094 033366 104460 ERROR ;ELSE, REPORT IT
6095 033370 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033370 104410 ; TRAP C$ESCAPE
033372 002520 .WORD L10073-.
6096 033374 GEDF EM50N,ERR50 ;IT WASN'T! SOMETHING WENT WRONG! REPORT IT
; "DEVICE FATAL" ERROR # 65

```

TEST 27 VIA TIMER # 1 ONE SHOT MODE

SEQ 0170

```

033374 104455
033376 000101
033400 017054
033402 010762
6097
6098 033404 344: ENDSUB
033404
033404 104403
6099
6100 ; .....
6101 ; TEST TIMER # 1 USING ONE SHOT MODE WITH OUTPUT ON PB7 ENABLED.
6102
6103 033406 BGNSUB
033406
033406 104402
6104 033410 004737 003762 JSR PC,MSTCLR ;INIT DMV & ENTER M LOOP
6105 033414 103003 BCC 1$ ;IF NO ERROR, PROCEED WITH TESTING
6106 033416 104460 ERROR ;ELSE, REPORT ERROR
6107 033420 ESCAPE TST ; & EXIT TEST
033420 104410
033422 002470
6108 033424 004537 004660 18: JSR R5,INIT1 ;INITIALIZE TIMER # 1
6109 033430 000000 0 ; 0 ==> LATCHES
6110 033432 000200 BIT7 ; MODE 2 & "T1" INT. ENABLE FLAG CLEARED
6111 033434 103003 BCC .+10 ;IF NO ERROR, PROCEED
6112 033436 104460 ERROR ;ELSE, REPORT IT
6113 033440 ESCAPE TST ; AND EXIT THIS TEST
033440 104410
033442 002450
6114
6115 ; MODE 2 IS ONE SHOT MODE WITH OUTPUT ON PB7 CONTROLLED BY TIMER 1
6116
6117 033444 004737 036146 JSR PC,GETT1 ;IS "T1" SET?
6118 033450 102002 BVC .+6 ;IF NO ERROR, PROCEED
6119 033452 104410 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
033454 002434
6120 033456 103123 BCC 6$ ;NO, GOOD.
6121 ;YES, REPORT IT'S NOT BEING CLEARED @ INIT.
6122 033460 004537 004064 JSR R5,READ ;GET ACR FOR ERROR MESSAGE
6123 033464 120013 ACR
6124 033466 002466 TMPB
6125 033470 103003 BCC .+10 ;IF NO ERROR, PROCEED
6126 033472 104460 ERROR ;ELSE, REPORT IT
6127 033474 ESCAPE TST ; AND EXIT THIS TEST
033474 104410
033476 002414
6128 033500 004537 004064 JSR R5,READ ;GET T1CL FOR ERROR MESSAGE
6129 033504 120004 T1CL
6130 033506 002450 TMP4
6131 033510 103003 BCC .+10 ;IF NO ERROR, PROCEED
6132 033512 104460 ERROR ;ELSE, REPORT IT
6133 033514 ESCAPE TST ; AND EXIT THIS TEST

```

```

TRAP C$ERDF
.WORD 65
.WORD EMSON
.WORD ERR50
L10074:
TRAP C$ESUB
T27.2:
TRAP C$BSUB
TRAP C$ERRCR
TRAP C$ESCAPE
.WORD L10073
TRAP C$ERROR
TRAP C$ESCAPE
.WORD L10073
TRAP C$ESCAPE
.WORD L10075
TRAP C$ERROR
TRAP C$ESCAPE
.WORD L10073
TRAP C$ERROR

```

TEST 27 VIA TIMER # 1 ONE SHOT MODE

SEQ 0171

```

033514 104410
033516 002374
6134 033520 004537 004064 JSR R5,READ ;GET T1CH FOR ERROR MESSAGE TRAP C$ESCAPE
6135 033524 120005 T1CH .WORD L10073 .
6136 033526 002452 TMP5
6137 033530 103003 BCC .+10 ;IF NO ERROR, PROCEED
6138 033532 104460 ERROR ;ELSE, REPORT IT
6139 033534 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033536 002354 TRAP C$ESCAPE
6140 033540 004537 004064 JSR R5,READ ;GET T1LL FOR ERROR MESSAGE .WORD L10073 .
6141 033544 120006 T1LL
6142 033546 002454 TMP6
6143 033550 103003 BCC .+10 ;IF NO ERROR, PROCEED
6144 033552 104460 ERROR ;ELSE, REPORT IT
6145 033554 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033556 002334 TRAP C$ESCAPE
6146 033560 004537 004064 JSR R5,READ ;GET T1LM FOR ERROR MESSAGE .WORD L10073 .
6147 033564 120007 T1LM
6148 033566 002456 TMP7
6149 033570 103003 BCC .+10 ;IF NO ERROR, PROCEED
6150 033572 104460 ERROR ;ELSE, REPORT IT
6151 033574 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033576 002314 TRAP C$ESCAPE
6152 033600 104455 GEDF EM50A,ERR50 ;REPORT "T1" NOT CLEARED @ INIT. .WORD L10073-.
033602 000102 ; "DEVICE FATAL" ERROR # 66
033604 016067 TRAP C$ERDF
033606 010762 .WORD 66
6153 .WORD EM50A
6154 .WORD ERR50
6155
6156 033610 112737 000002 002453 MOVB #2,TMP5.1
6157 033616 004537 004310 JSR R5,WRITE ;INIT TIMER # 1 BY WRITING INTO
6158 033622 120005 T1CH ;T1C-H (ADDR 05)
6159 033624 002453 TMP5.1
6160 033626 103003 BCC .+10 ;IF NO ERROR, PROCEED
6161 033630 104460 ERROR ;ELSE, REPORT IT
6162 033632 104410 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
033634 002256 TRAP C$ESCAPE
6163 033636 004737 036146 JSR PC,GETT1 ;IS "T1" SET? .WORD L10073 .
6164 033642 102002 BVC .+6 ;IF NO ERROR, PROCEED
6165 033644 104410 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT TRAP C$ESCAPE
033646 002242 .WORD L10075 .
6166 033650 103026 BCC 64 ;NO, GOOD.
6167 ;YES, REPORT IT'S NOT BEING CLEARED @ INIT.
6168 033652 004537 004064 JSR R5,READ ;GET T1CH FOR ERROR MESSAGE
6169 033656 120005 T1CH

```

TEST 27 - VIA TIMER # 1 ONE SHOT MODE

```

6170 033660 002452          TMP5
6171 033662 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6172 033664          104460          ERROR          ;ELSE, REPORT IT
6173 033666          104410          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
        033670 002222          ;          TRAP      C$ESCAPE
6174 033672 004537 004064          JSR      R5,READ          ;GET T1LM FOR ERROR MESSAGE          .WORD    L10073-.
6175 033676 120007          T1LM
6176 033700 002456          TMP7
6177 033702 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6178 033704          104460          ERROR          ;ELSE, REPORT IT
6179 033706          104410          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
        033710 002202          ;          TRAP      C$ESCAPE
6180 033712          GEDF      EM50B,ERR50          ;REPORT "T1" NOT CLEARED @ INIT.          .WORD    L10073-.
        033712 104455          ;          "DEVICE FATAL" ERROR # 67
        033714 000103          ;          TRAP      C$ERDF
        033716 016135          ;          .WORD    67
        033720 010762          ;          .WORD    EM50B
6181 033722          ESCAPE SUB          ;AND EXIT SUBTEST          .WORD    ERN50
        033722 104410          ;          TRAP      C$ESCAPE
        033724 002164          ;          .WORD    L10075-.
6182
6183
6184
6185 033726 004737 003762          61: JSR      PC,MSTCLR          ;INIT DMV & ENTER M-LOOP AGAIN
6186 033732 112737 000377 002445          MOVB     #377,TMP2+1          ;INITIAL VALUE FOR DDRB
6187 033740 004537 004310          JSR      R5,WRITE          ;LOAD IT
6188 033744 120002          DDRB
6189 033746 002445          TMP2+1
6190 033750 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6191 033752          104460          ERROR          ;ELSE, REPORT IT
6192 033754          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
        033754 104410          ;          TRAP      C$ESCAPE
        033756 002134          ;          .WORD    L10073 .
6193 033760 004537 004660          JSR      R5,INITT1          ;RE INITIALIZE THE TIMER
6194 033764 000000          O          ;          FOR MAXIMUM TIMEOUT
6195 033766 000200          BIT7          ;          MODE 2 & CLEARED "T1" INT. FLAG
6196 033770 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6197 033772          104460          ERROR          ;ELSE, REPORT IT
6198 033774          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
        033774 104410          ;          TRAP      C$ESCAPE
        033776 002114          ;          .WORD    L10073 .
6199 034000 004537 004064          JSR      R5,READ          ;GET ACR FOR FUTURE ERROR MESSAGES
6200 034004 120013          ACR
6201 034006 002466          TMPB
6202 034010 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6203 034012          104460          ERROR          ;ELSE, REPORT IT
6204 034014          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
        034014 104410          ;          TRAP      C$ESCAPE
        034016 002074          ;          .WORD    L10073 .

```

TEST 27 -- VIA TIMER # 1 ONE-SHOT MODE

6205	034020	004537	036114		JSR	R5,LODT1C		;LOAD TIMER # 1		
6206	034024	252		7#:	.BYTE	252				
6207	034025	252		8#:	.BYTE	252				
6208										
6209										
6210										
6211	034026	004737	036332		JSR	PC,GETPB7		;GET "PB7". IS IT CLEARED?		
6212	034032	102002			BVC	.+6		;IF NO ERROR, PROCEED		
6213	034034				ESCAPE	SUB		;ELSE, IT'S ALREADY BEEN REPORTED	EXIT	
	034034	104410							TRAP	C\$ESCAPE
	034036	002052							.WORD	L10075 .
6214	034040	103070			BCC	9#		;IT IS, GOOD		
6215	034042	004537	004064		JSR	R5,READ		;GET IFR FOR ERROR MESSAGE		
6216	034046	120015			IFR					
6217	034050	002472			TMPD					
6218	034052	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6219	034054				ERROR			;ELSE, REPORT IT		
	034054	104460							TRAP	C\$ERROR
6220	034056				ESCAPE	TST		; AND EXIT THIS TEST		
	034056	104410							TRAP	C\$ESCAPE
	034060	002032							.WORD	L10073 .
6221	034062	004537	004064		JSR	R5,READ		;GET T1CL FOR ERROR MESSAGE		
6222	034066	120004			T1CL					
6223	034070	002450			TMP4					
6224	034072	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6225	034074				ERROR			;ELSE, REPORT IT		
	034074	104460							TRAP	C\$ERROR
6226	034076				ESCAPE	TST		; AND EXIT THIS TEST		
	034076	104410							TRAP	C\$ESCAPE
	034100	002012							.WORD	L10073-.
6227	034102	004537	004064		JSR	R5,READ		;GET T1CH FOR ERROR MESSAGE		
6228	034106	120005			T1CH					
6229	034110	002452			TMP5					
6230	034112	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6231	034114				ERROR			;ELSE, REPORT IT		
	034114	104460							TRAP	C\$ERROR
6232	034116				ESCAPE	TST		; AND EXIT THIS TEST		
	034116	104410							TRAP	C\$ESCAPE
	034120	001772							.WORD	L10073 .
6233	034122	004537	004064		JSR	R5,READ		;GET T1LL FOR ERROR MESSAGE		
6234	034126	120006			T1LL					
6235	034130	002454			TMP6					
6236	034132	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6237	034134				ERROR			;ELSE, REPORT IT		
	034134	104460							TRAP	C\$ERROR
6238	034136				ESCAPE	TST		; AND EXIT THIS TEST		
	034136	104410							TRAP	C\$ESCAPE
	034140	001752							.WORD	L10073 .
6239	034142	004537	004064		JSR	R5,READ		;GET T1LH FOR ERROR MESSAGE		
6240	034146	120007			T1LH					
6241	034150	002456			TMP7					
6242	034152	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6243	034154				ERROR			;ELSE, REPORT IT		
	034154	104460							TRAP	C\$ERROR
6244	034156				ESCAPE	TST		; AND EXIT THIS TEST		
	034156	104410							TRAP	C\$ESCAPE
	034160	001732							.WORD	L10073-.

TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

```

6245 034162          GEDF   EM50V,ERR50   ;NO, STILL(?) SET!
;          "DEVICE FATAL" ERROR # 68
;          TRAP      C$ERDF
;          .WORD    68
;          .WORD    EM50V
;          .WORD    ERR50
        034162 104455
        034164 000104
        034166 017240
        034170 010762
6246 034172 004737 036204   JSR    PC,KICKT1   ;BECAUSE THE ERROR MESSAGE TAKES SO LONG TO
6247 034176 103003         BCC    .+10       ;IF NO ERROR, PROCEED
6248 034200          ERROR          ;ELSE, REPORT IT
        034200 104460
        034202          ESCAPE  TST          ;          AND EXIT THIS TEST
        034202 104410          TRAP      C$ERROR
        034204 001706          .WORD    C$ESCAPE
;          .WORD    L10073 .
6250 034206 004737 005032   JSR    PC,STALL   ; PROCESS & PRINT, RE START THE TIMER AND THEN
6251 034212 004737 005032   JSR    PC,STALL   ; DELAY FOR A LITTLE WHILE SO IT CAN DECREMENT
6252 034216 012703 000100   MOV    #100,R3    ; # INIT. "REPEAT" VALUE
6253 034222 004537 004064   JSR    R5,READ    ;READ THE LOW COUNTER
;          10$:
;          9$:
        034222 120004
        034226 002450
        034230 103003
        034232          BCC    .+10       ;IF NO ERROR, PROCEED
6256 034232 103003         ERROR          ;ELSE, REPORT IT
        034234 104460          TRAP      C$ERROR
6258 034236          ESCAPE  TST          ;          AND EXIT THIS TEST
        034236 104410          TRAP      C$ESCAPE
        034240 001652          .WORD    L10073 .
6259 034242 123737 002450 034024 CMPB   TMP4,7$    ;MAKE SURE THE COUNTER IS DECREMENTING
6260 034250 001013         BNE    12$        ;IT IS, NOW SEE IF THE HIGH COUNTER IS TOO
6261 034252 077315         SOB    R3,9$     ; # NO: IF NOT 64. ATTEMPTS, TRY AGAIN
6262 034254          GEDF   EM50D,ERR50   ;IT WASN'T -- REPORT THE ERROR
;          "DEVICE FATAL" ERROR # 69
;          TRAP      C$ERDF
;          .WORD    69
;          .WORD    EM50D
;          .WORD    ERR50
        034254 104455
        034256 000105
        034260 016251
        034262 010762
6263 034264 004737 036204   JSR    PC,KICKT1   ;RESTART TIMER AGAIN IF ERROR MESSAGE PRINTED
6264 034270 103003         BCC    12$        ;IF NO ERROR, PROCEED
6265 034272          ERROR          ;ELSE, REPORT IT
        034272 104460          TRAP      C$ERROR
6266 034274          ESCAPE  TST          ;          AND EXIT THIS TEST
        034274 104410          TRAP      C$ESCAPE
        034276 001614          .WORD    L10073 .
6267 034300 012703 000100   MOV    #100,R3    ;INIT. TIMEOUT VALUE
;          12$:
;          13$:
6268 034304 004537 004064   JSR    R5,READ    ;READ THE HIGH COUNTER
        034310 120005
        034312 002452
        034314 103003
        034316          BCC    .+10       ;IF NO ERROR, PROCEED
6272 034316 104460         ERROR          ;ELSE, REPORT IT
        034320          ESCAPE  TST          ;          AND EXIT THIS TEST
        034320 104410          TRAP      C$ERROR
        034322 001570          TRAP      C$ESCAPE
;          .WORD    L10073 .
6274 034324 123737 002452 034025 CMPB   TMP5,8$    ;DID IT CHANGE FROM THE LOADED VALUE?
6275 034332 001027         BNE    17$        ;YES, PROCEED WITH TESTING
6276 034334 077315         SOB    R3,13$     ;NO, IF NO TIMEOUT, TRY AGAIN
6277 034336 004537 004064   JSR    R5,READ    ;GET TILL FOR ERROR MESSAGE
6278 034342 120006
6279 034344 002454         TILL  TMP6

```

TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

```

6280 034346 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6281 034350                ERROR          ;ELSE, REPORT IT
6282 034350 104460          ESCAPE  TST          ;           AND EXIT THIS TEST          TRAP    C$ERROR
        034352 104410                .WORD    L10073 .
        034354 001536                .WORD
6283 034356 004537 004064    JSR      R5,READ          ;GET T1LH FOR ERROR MESSAGE
6284 034362 120007          T1LH
6285 034364 002456          TMP7
6286 034366 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6287 034370                ERROR          ;ELSE, REPORT IT
6288 034370 104460          ESCAPE  TST          ;           AND EXIT THIS TEST          TRAP    C$ERROR
        034372 104410                .WORD    C$ESCAPE
        034374 001516                .WORD    L10073 .
6289 034376                GEDF    EM50E,ERR50      ;ELSE, REPORT THAT HIGH COUNTER ISN'T RUNNING
        ;           "DEVICE FATAL" ERROR # 70
        034376 104455                .WORD    TRAP    C$ERDF
        034400 000106                .WORD    70
        034402 016305                .WORD    EM50E
        034404 010762                .WORD    ERR50
6290 034406                ESCAPE  SUB          ;IN THAT CASE, WE CAN'T PROCEED WITH TESTING EITHER
        034406 104410                .WORD    TRAP    C$ESCAPE
        034410 001500                .WORD    L10075-.
6291 ;-----
6292 ;-----
6293 034412 112737 000377 002445 17$:  MOVB    #377,TMP2+1      ;SETUP DDRB FOR DESIRED DIRECTION OF ORB
6294 034420 004537 004310          JSR      R5,WRITE
6295 034424 120002          DDRB
6296 034426 002445          TMP2+1
6297 034430 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6298 034432                ERROR          ;ELSE, REPORT IT
6299 034434 104460          ESCAPE  TST          ;           AND EXIT THIS TEST          TRAP    C$ERROR
        034434 104410                .WORD    TRAP    C$ESCAPE
        034436 001454                .WORD    L10073-.
6300 034440 004537 036114    JSR      R5,LODT1C      ;RE-LOAD TIMER # 1 WITH A VALUE WHICH WILL
6301 034444 001          18$:  .BYTE    1          ; CAUSE AN ALMOST IMMEDIATE TIMEOUT
6302 034445 000          19$:  .BYTE    0          ; (ADDRESS OF HIGH BYTE FOR T1C-H (ADDR 05))
6303 ;-----
6304 ;-----
6305 034446 004737 036146    JSR      PC,GETT1      ;WAS "T1" SET BY THE ABOVE OPERATION?
6306 034452 102002          BVC     .+6          ;IF NO ERROR, PROCEED
6307 034454                ESCAPE  SUB          ;ELSE, IT'S ALREADY BEEN REPORTED          EXIT
        034454 104410                .WORD    TRAP    C$ESCAPE
        034456 001432                .WORD    L10075 .
6308 034460 103446          BCS     20$          ;YES, OK - CONTINUE ERROR CHECKING
6309 034462 004537 004064    JSR      R5,READ          ;GET T1CL FOR ERROR MESSAGE
6310 034466 120004          T1CL
6311 034470 002450          TMP4
6312 034472 103003          BCC      .+10          ;IF NO ERROR, PROCEED
6313 034474                ERROR          ;ELSE, REPORT IT
6314 034474 104460          ESCAPE  TST          ;           AND EXIT THIS TEST          TRAP    C$ERROR
        034476 104410                .WORD    TRAP    C$ESCAPE
        034500 001412                .WORD    L10073 .
6315 034502 004537 004064    JSR      R5,READ          ;GET T1CH FOR ERROR MESSAGE

```


TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

6316	034506	120005			T1CH					
6317	034510	002452			TMP5					
6318	034512	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6319	034514				ERROR			;ELSE, REPORT IT		
	034514	104460							TRAP	C\$ERROR
6320	034516				ESCAPE	TST		; AND EXIT THIS TEST		
	034516	104410							TRAP	C\$ESCAPE
	034520	001372							.WORD	L10073 .
6321	034522	004537	004064		JSR	R5,READ		;GET T1LL FOR ERROR MESSAGE		
6322	034526	120006			T1LL					
6323	034530	002454			TMP6					
6324	034532	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6325	034534				ERROR			;ELSE, REPORT IT		
	034534	104460							TRAP	C\$ERROR
6326	034536				ESCAPE	TST		; AND EXIT THIS TEST		
	034536	104410							TRAP	C\$ESCAPE
	034540	001352							.WORD	L10073 .
6327	034542	004537	004064		JSR	R5,READ		;GET T1LH FOR ERROR MESSAGE		
6328	034546	120007			T1LH					
6329	034550	002456			TMP7					
6330	034552	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6331	034554				ERROR			;ELSE, REPORT IT		
	034554	104460							TRAP	C\$ERROR
6332	034556				ESCAPE	TST		; AND EXIT THIS TEST		
	034556	104410							TRAP	C\$ESCAPE
	034560	001332							.WORD	L10073-.
6333	034562				GEDF	EM50F,ERR50		;NO, BAD NEWS! REPORT THE FAILURE		
								; "DEVICE FATAL" ERROR # 71		
	034562	104455							TRAP	C\$ERDF
	034564	000107							.WORD	71
	034566	016341							.WORD	EM50F
	034570	10762							.WORD	ERR50
6334	034572				ESCAPE	SUB		; AND GET OUT OF SUBTEST		
	034572	104410							TRAP	C\$ESCAPE
	034574	001314							.WORD	L10075-.
6335	034576	004737	036332	20\$:	JSR	PC,GETPB7		;GET "PB7". IS IT SET?		
6336	034602	102002			BVC	.+6		;IF NO ERROR, PROCEED		
6337	034604				ESCAPE	SUB		;ELSE, IT'S ALREADY BEEN REPORTED - EXIT		
	034604	104410							TRAP	C\$ESCAPE
	034606	001302							.WORD	L10075 .
6338	034610	103445			BCS	41\$;YES, CONTINUE CHECKING 'T1"		
6339	034612	004537	004064		JSR	R5,READ		;GET T1CL FOR ERROR MESSAGE		
6340	034616	120004			T1CL					
6341	034620	002450			TMP4					
6342	034622	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6343	034624				ERROR			;ELSE, REPORT IT		
	034624	104460							TRAP	C\$ERROR
6344	034626				ESCAPE	TST		; AND EXIT THIS TEST		
	034626	104410							TRAP	C\$ESCAPE
	034630	001262							.WORD	L10073-.
6345	034632	004537	004064		JSR	R5,READ		;GET T1CH FOR ERROR MESSAGE		
6346	034636	120005			T1CH					
6347	034640	002452			TMP5					
6348	034642	103003			BCC	.+10		;IF NO ERROR, PROCEED		
6349	034644				ERROR			;ELSE, REPORT IT		
	034644	104460							TRAP	C\$ERROR
6350	034646				ESCAPE	TST		; AND EXIT THIS TEST		

TEST 27 -- VIA TIMER # 1 ONE-SHOT MODE

```

034646 104410 TRAP C$ESCAPE
034650 001242 .WORD L10073-.
6351 034652 004537 004064 JSR R5,READ ;GET T1LL FOR ERROR MESSAGE
6352 034656 120006 T1LL
6353 034660 002454 TMP6
6354 034662 103003 BCC .+10 ;IF NO ERROR, PROCEED
6355 034664 ERROR ;ELSE, REPORT IT
034664 104460 TRAP C$ERROR
6356 034666 ESCAPE TST ; AND EXIT THIS TEST
034666 104410 TRAP C$ESCAPE
034670 001222 .WORD L10073-.
6357 034672 004537 004064 JSR R5,READ ;GET T1LM FOR ERROR MESSAGE
6358 034676 120007 T1LM
6359 034700 002456 TMP7
6360 034702 103003 BCC .+10 ;IF NO ERROR, PROCEED
6361 034704 ERROR ;ELSE, REPORT IT
034704 104460 TRAP C$ERROR
6362 034706 ESCAPE TST ; AND EXIT THIS TEST
034706 104410 TRAP C$ESCAPE
034710 001202 .WORD L10073-.
6363 034712 GEDF EM50S,ERR50 ;NO! REPORT THAT PB7 DIDN'T GET SET!
; "DEVICE FATAL" ERROR # 72
034712 104455 TRAP C$ERDF
034714 000110 .WORD 72
034716 017122 .WORD EM50S
034720 010762 .WORD ERR50
6364 034722 000562 004064 BR 28$ ; & EXIT THIS SECTION OF SUBTEST
6365 034724 004537 004064 41$: JSR R5,READ ;READ T1C-H (ADDR 05) TO SEE IF IT CLEARS "T1"
6366 034730 120005 T1CH ;(THIS VALUE ISN'T CHECKED BECAUSE IT CAN BE
6367 034732 002452 TMP5 ;ALMOST ANYTHING)
6368 034734 103003 BCC .+10 ;IF NO ERROR, PROCEED
6369 034736 ERROR ;ELSE, REPORT IT
034736 104460 TRAP C$ERROR
6370 034740 ESCAPE TST ; AND EXIT THIS TEST
034740 104410 TRAP C$ESCAPE
034742 001150 .WORD L10073 .
6371 034744 004737 036146 JSR PC,GETT1 ;PUT THE CURRENT "T1" VALUE INTO THE CARRY BIT
6372 034750 102002 BVC 11$ ;IF NO ERROR, PROCEED
6373 034752 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
034752 104410 TRAP C$ESCAPE
034754 001134 .WORD L10075 .
6374 034756 103435 11$: BCS 21$ ;IF SET, ALL'S OK
6375 ;IF CLEARED! BAD VIA CHIP!
6376 034760 004537 004064 JSR R5,READ ;GET T1CL FOR ERROR MESSAGE
6377 034764 120004 T1CL
6378 034766 002450 TMP4
6379 034770 103003 BCC .+10 ;IF NO ERROR, PROCEED
6380 034772 ERROR ;ELSE, REPORT IT
034772 104460 TRAP C$ERROR
6381 034774 ESCAPE TST ; AND EXIT THIS TEST
034774 104410 TRAP C$ESCAPE
034776 001114 .WORD L10073 .
6382 035000 004537 004064 JSR R5,READ ;GET T1LL FOR ERROR MESSAGE
6383 035004 120006 T1LL
6384 035006 002454 TMP6
6385 035010 103003 BCC .+10 ;IF NO ERROR, PROCEED
6386 035012 ERROR ;ELSE, REPORT IT

```

TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

```

035012 104460
6387 035014          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP C$ERROR
      035014 104410
      035016 001074          .WORD          L10073-.
6388 035020 004537 004064 JSR      R5,READ          ;GET T1LM FOR ERROR MESSAGE
6389 035024 120007          T1LM
6390 035026 002456          TMP7
6391 035030 103003          BCC     .+10          ;IF NO ERROR, PROCEED
6392 035032          ERROR          ;ELSE, REPORT IT
      035032 104460
6393 035034          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP C$ERROR
      035034 104410          .WORD          L10073 .
      035036 001054
6394 035040          GEDF   EM50G,ERR50      ;REPORT BAD VIA CHIP!
      ;          "DEVICE FATAL" ERROR # 73
      TRAP C$ERDF
      .WORD 73
      .WORD EM50G
      .WORD ERR50
      035040 104455
      035042 000111
      035044 016406
      035046 010762
6395 035050 000507          BR      28$          ;BYPASS THE REST OF THIS SECTION OF TESTING
6396
6397 035052 004537 004064 21$: JSR      R5,READ          ;READ T1L-L (ADDR 06)
6398 035056 120006          T1LL
6399 035060 002454          TMP6          ;THIS SHOULD RETURN A 001
6400 035062 103003          BCC     .+10          ;IF NO ERROR, PROCEED
6401 035064          ERROR          ;ELSE, REPORT IT
      035064 104460          TRAP C$ERROR
6402 035066          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP C$ESCAPE
      035066 104410          .WORD          L10073 .
      035070 001022
6403 035072 123737 002454 034444 CMPB    TMP6,18$      ;CHECK T1L-L (ADDR 06) AGAINST LOADED VALUE
6404 035100 001415          BEQ     23$          ;IF SAME, PROCEED
6405
6406 035102 004537 004064 JSR      R5,READ          ;ELSE, REPORT BAD LOAD OF T1L-L (ADDR 06)
6407 035106 120007          T1LM          ;GET T1LM FOR ERROR MESSAGE
6408 035110 002456          TMP7
6409 035112 103003          BCC     .+10          ;IF NO ERROR, PROCEED
6410 035114          ERROR          ;ELSE, REPORT IT
      035114 104460          TRAP C$ERROR
6411 035116          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP C$ESCAPE
      035116 104410          .WORD          L10073 .
      035120 000772
6412 035122          GEDF   EM50H,ERR50      ;ELSE, REPORT BAD LOAD OF T1L-L (ADDR 06)
      ;          "DEVICE FATAL" ERROR # 74
      TRAP C$ERDF
      .WORD 74
      .WORD EM50H
      .WORD ERR50
      035122 104455
      035124 000112
      035126 016450
      035130 010762
6413 035132 000456          BR      28$          ;BYPASS THE REST OF THIS SECTION OF TESTING
6414
6415 035134 004737 036146 23$: JSR      PC,GETT1        ;IS "T1" STILL SET?
6416 035140 102002          BVC     .+6          ;IF NO ERROR, PROCEED
6417 035142          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
      TRAP C$ESCAPE
      .WORD L10075 .
      035142 104410
      035144 000744
6418 035146 103415          BCS    24$          ;YES, ALL'S OK
6419          ;NO! BAD VIA CHIP!

```

TEST 27 VIA TIMER # 1 ONE SHOT MODE

```

6420 035150 004537 004064      JSR    R5,READ      ;GET T1LH FOR ERROR MESSAGE
6421 035154 120007              T1LH
6422 035156 002456              TMP7
6423 035160 103003              BCC    .+10        ;IF NO ERROR, PROCEED
6424 035162 104460              ERROR          ;ELSE, REPORT IT
6425 035164 104410              ESCAPE TST        ;      AND EXIT THIS TEST          TRAP    C$ERROR
6426 035166 000724              GEDF    EM50I,ERR50 ;REPORT BAD VIA CHIP!          TRAP    C$ESCAPE
6427 035170 104455              ;      "DEVICE FATAL" ERROR # 75 .WORD  L10073 .
6428 035172 000113              ;                                TRAP    C$ERDF
6429 035174 016536              ;                                .WORD  75
6430 035176 010762              ;                                .WORD  EM50I
6431 035200 000433              BR      28$        ;BYPASS THE REST OF THIS SECTION OF TESTING .WORD  ERR50
6432 035202 004537 004064      24$: JSR    R5,READ      ;READ T1L-H (ADDR 07)
6433 035206 120007              T1LH
6434 035210 002456              TMP7
6435 035212 103003              BCC    .+10        ;THIS SHOULD RETURN A 000
6436 035214 104460              ERROR          ;IF NO ERROR, PROCEED
6437 035216 104410              FSCAPE TST        ;      AND EXIT THIS TEST          TRAP    C$ERROR
6438 035220 000672              ;                                TRAP    C$ESCAPE
6439 035222 123737 002456 034445 CMPB    TMP7,19$   ;CHECK T1L-H (ADDR 07) AGAINST LOADED VALUE .WORD  L10073-.
6440 035230 001405              BEQ    26$
6441 035232 104455              GEDF    EM50J,ERR50 ;ELSE, REPORT BAD LOAD OF T1L-H (ADDR 07)
6442 035234 000114              ;      "DEVICE FATAL" ERROR # 76          TRAP    C$ERDF
6443 035236 016600              ;                                .WORD  76
6444 035240 010762              ;                                .WORD  EM50J
6445 035242 000412              BR      28$        ;BYPASS THE REST OF THIS SECTION OF TESTING .WORD  ERR50
6446 035244 004737 036146      26$: JSR    PC,GETT1   ;IS "T1" STILL SET?
6447 035250 102002              BVC    .+6
6448 035252 104410              ESCAPE SUB        ;IF NO ERROR, PROCEED
6449 035254 000634              ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT          TRAP    C$ESCAPE
6450 035256 103404              BCS    28$        ;YES, ALL'S OK          .WORD  L10075 .
6451 035260 104455              GEDF    EM50K,ERR50 ;NO! BAD VIA CHIP!
6452 035262 000115              ;      "DEVICE FATAL" ERROR # 77          TRAP    C$ERDF
6453 035264 016666              ;                                .WORD  77
6454 035266 010762              ;                                .WORD  EM50K
6455 035270 004537 004064      28$: JSR    R5,READ      ;READ T1C L (ADDR 04) TO CLEAR "T1"
6456 035274 120004              T1CL
6457 035276 002450              TMP4
6458 035300 103003              BCC    .+10        ;(THIS VALUE ISN'T CHECKED BECAUSE IT CAN BE
6459 035302 104460              ERROR          ;ALMOST ANYTHING)
6460 035302 104460              ;IF NO ERROR, PROCEED
6461 035302 104460              ;ELSE, REPORT IT          TRAP    C$ERROR

```

TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

```

6453 035304          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP    C$ESCAPE
      035304 104410          ;          .WORD    L10073-.
      035306 000604
6454 035310 004737 036146    JSR    PC,GETT1      ;IS "T1" CLEARED NOW
6455 035314 102002          BVC    16$          ;IF NO ERROR, PROCEED
6456 035316          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
      035316 104410          ;          TRAP    C$ESCAPE
      035320 000570          ;          .WORD    L10075 .
6457 035322 103004          16$:  BCC    29$          ;YES, ALL'S OK
6458 035324          GEDF   EMS0C,ERR50 ;NO! BAD VIA CHIP!
      ;          "DEVICE FATAL" ERROR # 78
      ;          TRAP    C$ERDF
      ;          .WORD    78
      ;          .WORD    EMS0C
      ;          .WORD    ERR50
      ;
      ;-----
6459
6460
6461
6462 035334 105037 002445    29$:  CLRB   TMP2+1   ;CHANGE THE DIRECTION OF ORB - IT SHOULDN'T
6463 035340 004537 004310    JSR    R5,WRITE    ; HAVE ANY EFFECT ON "PB7"
6464 035344 120002          DDRB
6465 035346 002445          TMP2+1
6466 035350 103003          BCC    .+10        ;IF NO ERROR, PROCEED
6467 035352          ERROR          ;ELSE, REPORT IT
      035352 104460          ;          TRAP    C$ERROR
6468 035354          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP    C$ESCAPE
      035354 104410          ;          .WORD    L10073 .
      035356 000534
6469 035360 004537 004310    JSR    R5,WRITE    ;RE-WRITE INTO TIC-H (ADDR 05) TO SET T1 AGAIN
6470 035364 120005          T1CH
6471 035366 002453          TMP5+1
6472 035370 103003          BCC    .+10        ;IF NO ERROR, PROCEED
6473 035372          ERROR          ;ELSE, REPORT IT
      035372 104460          ;          TRAP    C$ERROR
6474 035374          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP    C$ESCAPE
      035374 104410          ;          .WORD    L10073-.
      035376 000514
6475 035400 004737 036146    JSR    PC,GETT1      ;IS 'T1" SET AGAIN
6476 035404 102002          BVC    .+6          ;IF NO ERROR, PROCEED
6477 035406          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
      035406 104410          ;          TRAP    C$ESCAPE
      035410 000500          ;          .WORD    L10075 .
6478 035412 103426          BCS    32$          ;YES, ALL'S WELL (AGAIN?)
6479 035414 004537 004064    JSR    R5,READ     ;GET T1CH FOR ERROR MESSAGE
6480 035420 120005          T1CH
6481 035422 002452          TMP5
6482 035424 103003          BCC    .+10        ;IF NO ERROR, PROCEED
6483 035426          ERROR          ;ELSE, REPORT IT
      035426 104460          ;          TRAP    C$ERROR
6484 035430          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP    C$ESCAPE
      035430 104410          ;          .WORD    L10073-.
      035432 000460
6485 035434 004537 004064    JSR    R5,READ     ;GET T1LH FOR ERROR MESSAGE
6486 035440 120007          T1LH
6487 035442 002456          TMP7
6488 035444 103003          BCC    .+10        ;IF NO ERROR, PROCEED
6489 035446          ERROR          ;ELSE, REPORT IT

```

TEST 27 - VIA TIMER # 1 ONE-SHOT MODE

```

035446 104460
6490 035450 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ERROR
035450 104410 TRAP C$ESCAPE
035452 000440 .WORD L10073
6491 035454 GEDF EMSOL,ERR50 ;NO! SOMETHING WENT WRONG! REPORT IT
; "DEVICE FATAL" ERROR # 79
035454 104455 TRAP C$ERDF
035456 000117 .WORD 79
035460 016730 .WORD EMSOL
035462 010762 .WORD ERR50
6492 035464 ESCAPE SUB ; AND EXIT FROM THIS SUBTEST
035464 104410 TRAP C$ESCAPE
035466 000422 .WORD L10075-.

-----
6493
6494
6495
6496 035470 004737 036332 32$: JSR PC,GETPB7 ;GET "PB7". IS IT SET?
6497 035474 102002 BVC .+6 ;IF NO ERROR, PROCEED
6498 035476 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
035476 104410 TRAP C$ESCAPE
035500 000410 .WORD L10075-.
6499 035502 103404 BCS 44$ ;YES, GOOD.
6500 035504 GEDF EMSOU,ERR50 ;NO, BAD! REPORT IT; NOT SET AFTER TIMEOUT
; "DEVICE FATAL" ERROR # 80
035504 104455 TRAP C$ERDF
035506 000120 .WORD 80
035510 017174 .WORD EMSOU
035512 010762 .WORD ERR50
6501 035514 112737 000125 002455 44$: MOVB #125,TMP6+1 ;USING A DIFFERENT VALUE -- 55 HEX.,
6502 035522 004537 004310 JSR R5,WRITE ;RE-LOAD T1L-L (ADDR 06)
6503 035526 120006 TILL
6504 035530 002455 TMP6+1
6505 035532 103003 BCC .+10 ;IF NO ERROR, PROCEED
6506 035534 ERROR ;ELSE, REPORT IT
035534 104460 TRAP C$ERROR
6507 035536 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ESCAPE
035536 104410 .WORD L10073-.
035540 000352
6508 035542 004737 036146 JSR PC,GETT1 ;IS "T1" STILL SET?
6509 035546 102002 BVC .+6 ;IF NO ERROR, PROCEED
6510 035550 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
035550 104410 TRAP C$ESCAPE
035552 000336 .WORD L10075-.
6511 035554 103416 BCS 33$ ;YES, ALL'S STILL OK
6512 035556 004537 004064 JSR R5,READ ;GET TILL FOR ERROR MESSAGE
6513 035562 120006 TILL
6514 035564 002454 TMP6
6515 035566 103003 BCC .+10 ;IF NO ERROR, PROCEED
6516 035570 ERROR ;ELSE, REPORT IT
035570 104460 TRAP C$ERROR
6517 035572 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ESCAPE
035572 104410 .WORD L10073
035574 000316
6518 035576 GEDF EMSOM,ERR50 ;NO! SOMETHING WENT WRONG! REPORT IT
; "DEVICE FATAL" ERROR # 81
035576 104455 TRAP C$ERDF
035600 000121 .WORD 81

```

TEST 27 -- VIA TIMER # 1 ONE-SHOT MODE

```

035602 017012
035604 010762
6519 035606          ESCAPE SUB          ;          AND EXIT FROM THIS SUBTEST
035606 104410
035610 000300
6520
6521
6522 ;-----
6523 035612 112737 000125 002453 33$: MOVB #125,TMP5+1 ;AND USING THE SAME VALUE AGAIN (55 HEX),
6524 035620 004537 004310 JSR R5,WRITE ;NOW LOAD TIC-H (ADDR 05)
6525 035624 120005 T1CH
6526 035626 002453 TMP5+1
6527 035630 103003 BCC .+10 ;IF NO ERROR, PROCEED
6528 035632 ERROR ;ELSE, REPORT IT
035632 104460
6529 035634          ESCAPE TST          ;          AND EXIT THIS TEST
035634 104410
035636 000254
6530 035640 004737 036146 JSR PC,GETT1 ;"T1" SHOULD NOW BE CLEARED
6531 035644 102002 BVC .+6 ;IF NO ERROR, PROCEED
6532 035646          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
035646 104410
035650 000240
5533 035652 103024 BCC 34$ ;IT WAS, ALL'S WELL THAT END'S WELL (I THINK!?)
6534 035654 004537 004064 JSR R5,READ ;GET T1L FOR ERROR MESSAGE
6535 035660 120006 T1LL
6536 035662 002454 TMP6
6537 035664 103003 BCC .+10 ;IF NO ERROR, PROCEED
6538 035666 ERROR ;ELSE, REPORT IT
035666 104460
6539 035670          ESCAPE TST          ;          AND EXIT THIS TEST
035670 104410
035672 000220
6540 035674 004537 004064 JSR R5,READ ;GET T1LH FOR ERROR MESSAGE
6541 035700 120007 T1LH
6542 035702 002456 TMP7
6543 035704 103003 BCC .+10 ;IF NO ERROR, PROCEED
6544 035706 ERROR ;ELSE, REPORT IT
035706 104460
6545 035710          ESCAPE TST          ;          AND EXIT THIS TEST
035710 104410
035712 000200
6546 035714          GEDF EM50N,ERR50 ;IT WASN'T! SOMETHING WENT WPONG! REPORT IT
; "DEVICE FATAL" ERROR # 82
035714 104455
035716 000122
035720 017054
035722 010762
6547
6548 035724 004537 004310 34$: JSR R5,WRITE ;RE LOAD TIC H (ADDR 5) TO START IT AGAIN
6549 035730 120005 T1CH
6550 035732 002453 TMP5+1
6551 035734 103003 BCC .+10 ;IF NO ERROR, PROCEED
6552 035736 ERROR ;ELSE, REPORT IT
035736 104460
6553 035740          ESCAPE TST          ;          AND EXIT THIS TEST
035740 104410

```


TEST 27 VIA TIMER # 1 ONE SHOT MODE

SEQ 0184

036106 010762
6589 036110
036110
036110 104403
6590 036112
036112
036112 104401

481: ENDSUB
ENDTST

.WORD ERR50
L10075: TRAP C\$ESUB
L10073: TRAP C\$ETST

6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603

: L0DT1C -- LOAD TIMER ONE AT ADDRESSES 04 & 05
: CALLING SEQUENCE:
: JSR R5,L0DT1C
: .BYTE <VALUE FOR T1L L (ADDRESS 04)>
: .BYTE <VALUE FOR T1C-H (ADDRESS 05)>
: <NEXT SEQUENTIAL INSTRUCTION

6604 036114 112537 002451
6605 036120 112537 002453
6606 036124 004537 004310
6607 036130 120004
6608 036132 002451
6609 036134 004537 004310
6610 036140 120005
6611 036142 002453
6612 036144 000205

L0DT1C: MOVB (R5),TMP4.1 ;SETUP TO LOAD T1CL
MOVB (R5),TMP5.1 ; AND T1CH
JSR R5,WRITE ;LOAD T1C-L (ADDR 04) WITH PASSED PARAMETER
T1CL
TMP4.1
JSR R5,WRITE ;LOAD T1C-H (ADDR 05) WITH PASSED PARAMETER
T1CH ; (THIS WILL ALSO RESET "T1" & THE COUNTER)
TMP5.1
RTS R5

6613
6614
6615
6616
6617
6618
6619

: GETT1 -- GET THE "T1" FLAG FROM THE VIA'S IFR REGISTER AND PUT IT
: INTO THE "CARRY" BIT

6620 036146 004537 004064
6621 036152 120015
6622 036154 002472
6623 036156 103003
6624 036160
036160 104460
6625 036162 000262
6626 036164 000207
6627

GETT1: JSR R5,READ ;GET VIA S IFR REG.
IFR
TMPD
BCC 1\$;IF NO ERROR, PROCEED
ERROR ;ELSE, REPORT IT
SEV ;FLAG AN ERROR TO MAINLINE ROUTINE TRAP C\$ERROR
RTS PC ; AND TAKE AN ABNORMAL RETURN

6628 036166 010046
6629 036170 113700 002472
6630 036174 106100
6631 036176 106100
6632 036200 012600
6633 036202 000207

1\$: MOV R0,(SP) ;PRESERVE R0
MOVB TMPD,R0 ;PUT VALUE HERE TO PRESERVE TMPD
ROLB R0 ;"IRQ" GOES INTO CARRY BIT
ROLB R0 ;"T1" GOES INTO CARRY BIT
MOV (SP),R0 ;RESTORE R0
RTS PC

6634
6635
6636
6637
6638
6639

: KICKT1 INIT. TIMER # 1 BY THE FOLLOWING PROCEDURE:
: READ T1L H (ADDR 07) TO GET THE LAST VALUE LOADED INTO IT

TEST 27 VIA TIMER # 1 ONE-SHOT MODE

```

6640 ; WRITE THAT VALUE INTO TIC-H (ADDR 05) TO RESET THE "T1" INTERRUPT FLAG
6641 ; AND CAUSE THE RE-LOADING OF BOTH COUNTERS.
6642 ;
6643 ;
6644 036204 010346 KICKT1: MOV R3,-(SP) ;SAVE CALLER'S REGISTER CONTENTS
6645 036206 004537 004064 JSR R5,READ ;GET THE CURRENT SETTING OF THE HIGH LATCH
6646 036212 120007 T1LH
6647 036214 002456 TMP7
6648 036216 103443 BCS 10$ ;IF ERROR, EXIT
6649
6650 036220 012777 120005 144134 MOV #T1CH,#SEL4 ;SETUP ADDRESS FOR M-LOOP WRITE
6651 036226 113777 002456 144132 MOVB TMP7,#SEL6 ;SETUP DATA FOR SAME
6652 036234 113737 002456 002453 MOVB TMP7,TMP5+1 ;PUT HERE TOO. BECAUSE WE'RE GOING TO WRITE IT.
6653 036242 142777 000100 144110 BICB #IFRT1,#SEL3 ;CLEAR THE INTERRUPT BIT -- JUST IN CASE
6654 036250 112777 000002 144100 MOVB #WRILOC,#SEL2 ;TELL THE M-LOOP TO WRITE THE BYTE FOR US
6655 036256 012703 000074 MOV #60,R3 ;SETUP TIMEOUT COUNTER
6656 036262 132777 000200 144066 5$: BITB #MRDY,#SEL2 ;WAIT FOR M READY TO BE SET
6657 036270 001016 BNE 10$ ;AS SOON AS "MRDY" IS SET, EXIT!
6658 036272 077305 SOB R3,5$ ;IF NO TIMEOUT, CHECK AGAIN FOR M-READY
6659 036274 GTDF EM4,ERR4 ;ELSE, "MRDY" TIMEOUT
; QUEUE "DEVICE FATAL" ERROR # 84
; MOV #T.EDF,ERRTYP
; MOV #84,ERRNBR
; MOV #EM4,ERRMSG
; MOV #ERR4,ERRBLK
036274 012737 000001 002236
036302 012737 000124 002240
036310 012737 014500 002242
036316 012737 005426 002244
6660 036324 000261 SEC ;INDICATE THE FAILURE & EXIT
6661
6662 036326 012603 10$: MOV (SP)+,R3 ;RESTORE REGISTER
6663 036330 000207 RTS PC ;IMMEDIATE RETURN
6664
6665 ;-----
6666 ; GETPB7 -- PUT THE CURRENT SETTING OF "PB7" (BIT 7 OF ORB W/IN THE VIA CHIP)
6667 ; INTO THE CARRY BIT SO IT CAN BE TESTED UPON RETURN.
6668 ;
6669 ; CALLING SEQUENCE:
6670 ;
6671 ; JSR PC,GETPB7
6672 ; <TEST FOR PB7 SET OR CLEARED WITH "BCS" OR "BCC" INSTR'S>
6673 ;-----
6674
6675 036332 004537 004064 GETPB7: JSR R5,READ ;GET THE REGISTER THAT CONTAINS "PB7"
6676 036336 120000 ORB
6677 036340 002440 TMO
6678 036342 103003 BCC 1$ ;IF NO ERROR, PROCEED
6679 036344 ERROR ;ELSE, REPORT IT
036344 104460 TRAP C$ERROR
6680 036346 000262 SEV ;FLAG AN ERROR TO MAINLINE ROUTINE
6681 036350 000207 RTS PC ; AND TAKE AN ABNORMAL RETURN
6682
6683 036352 010046 1$: MOV R0,(SP) ;PRESERVE THIS REGISTER FOR THE CALLER
6684 036354 113700 002440 MOVB TMO,R0 ;PUT ITS CONTENTS HERE SO WE CAN MANIPULATE IT
6685 036360 106100 ROLB R0 ;PUT "PB7" INTO THE CARRY BIT
6686 036362 012600 MOV (SP)+,R0 ;RESTORE R0 FOR THE CALLER
6687 036364 000207 RTS PC ;RETURN WITH "PB7" IN THE CARRY BIT
6688

```

TEST 28 VIA TIMER 1 FREE RUNNING MODE TEST

6733

.SBTTL TEST 28 - VIA TIMER 1 FREE-RUNNING MODE TEST

```

*****
:
: TEST 28 VIA TIMER 1 FREE-RUNNING MODE TEST
:
: THIS TEST VERIFIES THAT THE TIMER 1 COUNTER IS OPERATIONAL IN
: FREE-RUNNING MODE, IN EACH OF TWO SUBTESTS.
:
: THE PROGRAM PERIODICALLY CHECKS THE COUNTER TO VERIFY THAT:
:
: IT IS DECREMENTING AND EVENTUALLY REACHES 0,
:
: RELOADS FROM THE LATCHES, AND
:
: CONTINUES TO DECREMENT.
:
: IN THE FIRST SUBTEST, THE FOLLOWING IS PERFORMED :
:
: A MASTER CLEAR IS DONE AND THE TIMER IS PLACED IN FREE RUNNING MODE
: BY SETTING ACR7 TO 0 & ACR6 TO 1 (MODE 1). AND THE PROGRAM CHECKS
: FOR THE "T1" (BIT 6 IN IFR) TO BE INITIALLY CLEARED.
:
: THEN T1L L (ADR 04) IS LOADED WITH 125 (OCTAL) AND T1C H (ADR 05) IS
: LOADED WITH 125 (OCTAL) STARTING THE COUNTER.
:
: THE COUNT IS ALLOWED TO REACH 0 AGAIN, AND THE 'T1' IS READ AND
: CHECKED TO BE SET.
:
: T1C H (ADR 05) IS READ AND "T1" IS CHECKED TO BE STILL SET.
:
: THE COUNTER LO BYTE IS READ AND THE 'T1' IS READ AND CHECKED TO BE
: CLEARED BY THE READ OF T1C L.
:
: THE COUNT IS ALLOWED TO REACH 0 ONCE MORE AND "T1" IS CHECKED TO BE
: SET AGAIN.
:
: T1L L IS LOADED WITH 252 (OCTAL) AND "T1" IS CHECKED TO BE STILL
: SET.
:
: T1C H IS LOADED WITH 252 (OCTAL) AND "T1" IS READ AND CHECKED TO BE
: CLEARED BY THE LOADING OF T1C H.
:
: IN THE SECOND SUBTEST, ALL OF THE ABOVE OPERATIONS ARE REPEATED, WITH
: ACR7 = 1, AND ACR6 = 1 (MODE 3). ALSO, PB7 IS VERIFIED FOR PROPER
: STATE AT THE PROPER TIME.

```

```

036366
6734 036366
036366
036366 104402
6735 036370 004737 003762
6736 036374 103003
6737 036376

```

```

:
: BGNTST
:
: BGNSUB
:
: JSR PC,MSTCLR ;INIT DMV & ENTER M-LOOP
: BCC 1$ ;IF NO ERROR, PROCEED WITH TESTING
: ERROR ;ELSE, REPORT ERROR
:
: T28::
: T28.1: TRAP C$BSUB

```

TEST 28 VIA TIMER 1 FREE RUNNING MODE TEST

SEQ 0187

```

036376 104460
6738 036400          ESCAPE TST          ; & EXIT TEST          TRAP C$ERROR
036400 104410
036402 001514          TRAP C$ESCAPE
6739 036404 004537 004660 1$: JSR R5,INITT1      ;INITIALIZE TIMER # 1
6740 036410 000000          0 ==> LATCHES
6741 036412 000100          ; MODE 1 & "T1" INT. ENABLE FLAG CLEARED
6742 036414 103003          BCC .+10          ;IF NO ERROR, PROCEED
6743 036416          ERROR          ;ELSE, REPORT IT
036416 104460          TRAP C$ERROR
6744 036420          ESCAPE TST          ; AND EXIT THIS TEST    TRAP C$ERROR
036420 104410          TRAP C$ESCAPE
036422 001474          .WORD L10076 .
6745 036424 004737 036146 JSR PC,GETT1      ;IS "T1" SET?
6746 036430 102002          BVC .+6          ;IF NO ERROR, PROCEED
6747 036432          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
036432 104410          TRAP C$ESCAPE
036434 000414          .WORD L10077 .
6748 036436 103006          BCC 2$          ;NO, GOOD.
6749 036440          GEDF EM50A,ERR50 ;YES, REPORT IT'S NOT BEING CLEARED @ INIT.
          ; "DEVICE FATAL" ERROR # 85
          TRAP C$ERDF
          .WORD 85
          .WORD EM50A
          .WORD ERR50
036440 104455
036442 000125
036444 016067
036446 010762
6750 036450          ESCAPE SUB          ; & EXIT TEST
036450 104110          TRAP C$ESCAPE
036452 000376          .WORD L10077-.
6751
6752
6753
6754 036454 004537 036114 2$: JSR R5,LODT1C    ;RELOAD TIMER 1'S COUNTERS WITH NEW VALUES:
6755 036460 125 125          .BYTE 125,125
6756
6757
6758
6759 036462 005003          CLR R3          ;INITIALIZE TIMEOUT COUNTER
6760 036464 004737 036146 3$: JSR PC,GETT1      ;"T1" SHOULD BE SET. IS IT?
6761 036470 102002          BVC .+6          ;IF NO ERROR, PROCEED
6762 036472          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
036472 104410          TRAP C$ESCAPE
036474 000354          .WORD L10077-.
6763 036476 103407          BCS 4$          ;YES, GOOD.
6764 036500 077307          SOB R3,3$       ;NO, IF NO TIMEOUT, LOOK AGAIN
6765 036502          GEDF EM50F,ERR50 ;ELSE, SAY IT WASN'T SET BY T1 TIMEOUT
          ; "DEVICE FATAL" ERROR # 86
          TRAP C$ERDF
          .WORD 86
          .WORD EM50F
          .WORD ERR50
036502 104455
036504 000126
036506 016341
036510 010762
6766 036512          ESCAPE SUB          ;IF ERROR, THE REST OF THIS TEST IS UN-DOABLE!
036512 104410          TRAP C$ESCAPE
036514 000334          .WORD L10077 .
6767
6768 036516 004537 004064 4$: JSR R5,READ      ;READING T1CH SHOULDN'T CLEAR 'T1'
6769 036522 120005          T1CH
6770 036524 002452          TMP5
          ; (WE DON'T CARE WHAT THIS IS)

```

TEST 28 - VIA TIMER 1 FREE RUNNING MODE TEST

```

6771 036526 103003      BCC      .+10      ;IF NO ERROR, PROCEED
6772 036530      ERROR      ;ELSE, REPORT IT
        036530 104460      ESCAPE TST      ;           AND EXIT THIS TEST      TRAP      C$ERROR
6773 036532      ESCAPE TST      ;           AND EXIT THIS TEST      TRAP      C$ESCAPE
        036532 104410      ;           AND EXIT THIS TEST      .WORD    L10076 .
        036534 001362
6774 036536 004737 036146 JSR      PC,GETT1   ;CHECK "T1" -- IT SHOULD STILL BE SET
6775 036542 102002      BVC      .+6       ;IF NO ERROR, PROCEED
6776 036544      ESCAPE SUB      ;ELSE, IT'S ALREADY BEEN REPORTED  EXIT
        036544 104410      ;           AND EXIT THIS TEST      TRAP      C$ESCAPE
        036546 000302      ;           AND EXIT THIS TEST      .WORD    L10077 .
6777 036550 103404      BCS      6$       ;IT IS, GOOD.
6778 036552      GEDF     EMSOG,ERR50 ;CLEARED BY READING T1CH!!
        ;           "DEVICE FATAL" ERROR # 87
        ;           AND EXIT THIS TEST      TRAP      C$ERDF
        ;           AND EXIT THIS TEST      .WORD    87
        ;           AND EXIT THIS TEST      .WORD    EMSOG
        ;           AND EXIT THIS TEST      .WORD    ERR50
        036552 104455
        036554 000127
        036556 016406
        036560 010762
6779 036562 004737 036204 6$: JSR      PC,KICKT1  ;KICK IT OFF AGAIN SO WE CAN PRESERVE TIMING
6780 036566 103003      BCC      .+10     ;IF NO ERROR, PROCEED
6781 036570      ERROR      ;ELSE, REPORT IT
        036570 104460      ESCAPE TST      ;           AND EXIT THIS TEST      TRAP      C$ERROR
6782 036572      ESCAPE TST      ;           AND EXIT THIS TEST      TRAP      C$ESCAPE
        036572 104410      ;           AND EXIT THIS TEST      .WORD    L10076 .
        036574 001322
6783
6784 ;WAIT FOR IT TO FINISH:
6785 036576 005003      CLR      R3       ;INITIALIZE TIMEOUT COUNTER
6786 036600 004737 036146 7$: JSR      PC,GETT1   ;"T1" SHOULD BE SET. IS IT?
6787 036604 102002      BVC      .+6       ;IF NO ERROR, PROCEED
6788 036606      ESCAPE SUB      ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
        036606 104410      ;           AND EXIT THIS TEST      TRAP      C$ESCAPE
        036610 000240      ;           AND EXIT THIS TEST      .WORD    L10077-.
6789 036612 103402      BCS      8$       ;YES, GOOD.
6790 036614 077307      SOB      R3,7$    ;NO. IF NO TIMEOUT, LOOK AGAIN
6791 036616 000422      BR      10$      ;IF TIMEOUT, BYPASS NEXT CHECK (THIS DONE ABOVE)
6792 036620 004537 004064 8$: JSR      R5,READ   ;READING T1CL SHOULD CLEAR 'T1"
6793 036624 120004      T1CL
6794 036626 002450      TMP4
6795 036630 103003      BCC      .+10     ; (WE DON'T CARE WHAT THIS IS EITHER)
6796 036632      ERROR      ;IF NO ERROR, PROCEED
        036632 104460      ;ELSE, REPORT IT
        ;           AND EXIT THIS TEST      TRAP      C$ERROR
6797 036634      ESCAPE TST      ;           AND EXIT THIS TEST      TRAP      C$ESCAPE
        036634 104410      ;           AND EXIT THIS TEST      .WORD    L10076 .
        036636 001260
6798 036640 004737 036146 JSR      PC,GETT1   ;CHECK "T1" -- IT SHOULD BE CLEARED NOW
6799 036644 102002      BVC      .+6       ;IF NO ERROR, PROCEED
6800 036646      ESCAPE SUB      ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
        036646 104410      ;           AND EXIT THIS TEST      TRAP      C$ESCAPE
        036650 000200      ;           AND EXIT THIS TEST      .WORD    L10077-.
6801 036652 103004      BCC      10$      ;IT IS, GOOD.
6802 036654      GEDF     EMSOC,ERR50 ;NOT CLEARED! REPORT IT.
        ;           "DEVICE FATAL" ERROR # 88
        ;           AND EXIT THIS TEST      TRAP      C$ERDF
        ;           AND EXIT THIS TEST      .WORD    88
        ;           AND EXIT THIS TEST      .WORD    EMSOC
        ;           AND EXIT THIS TEST      .WORD    ERR50
        036654 104455
        036656 000130
        036660 016203
        036662 010762

```


TEST 28 - VIA TIMER 1 FREE-RUNNING MODE TEST

```

6834 037036 103004          BCC      18$      ;IT DID -- GOOD.
6835 037040          GEDF     EM50A,ERR50 ;NOP! REPORT: "T1" NOT CLEARED BY LOADING T1LH
;          "DEVICE FATAL" ERROR # 91
          037040 104455          TRAP     C$ERDF
          037042 000133          .WORD   91
          037044 016067          .WORD   EM50A
          037046 010762          .WORD   ERR50
6836 037050          18$:      ;THAT'S ALL FOLKS!
6837 037050          ENDSUB
          037050          L10077:
          037050 104403          TRAP     C$ESUB
6838          ;-----
6839 037052          BGNSUB
          037052          T28.2:
6840 037052 104402          TRAP     C$BSUB
6840 037054 004737 003762    JSR      PC,MSTCLR      ;INIT DMV & ENTER M-LOOP
6841 037060 103003          BCC      1$           ;IF NO ERROR, PROCEED WITH TESTING
6842 037062          ERROR          ;ELSE, REPORT ERROR
          037062 104460          TRAP     C$ERROR
6843 037064          ESCAPE   TST           ; & EXIT TEST
          037064 104410          TRAP     C$ESCAPE
          037066 001030          .WORD   L10076
6844 037070 112737 000377 002445 1$:  MOVB    #377,TMP2+1    ;SETUP DDRB SUCH THAT ORB IS AN INPUT/OUTPUT REG
6845 037076 004537 004310    JSR      R5,WRITE
6846 037102 120002          DDRB
6847 037104 002445          TMP2+1
6848 037106 103003          BCC      .+10        ;IF NO ERROR, PROCEED
6849 037110          ERROR          ;ELSE, REPORT IT
          037110 104460          TRAP     C$ERROR
6850 037112          ESCAPE   TST           ; AND EXIT THIS TEST
          037112 104410          TRAP     C$ESCAPE
          037114 001002          .WORD   L10076
6851 037116 112737 000030 002441    MOVB    #30,TMP0+1    ;CLEAR ALL BITS IN ORB EXCEPT DTR L & RTS L
6852 037124 004537 004310    JSR      R5,WRITE
6853 037130 120000          ORB
6854 037132 002441          TMP0+1
6855 037134 103003          BCC      .+10        ;IF NO ERROR, PROCEED
6856 037136          ERROR          ;ELSE, REPORT IT
          037136 104460          TRAP     C$ERROR
6857 037140          ESCAPE   TST           ; AND EXIT THIS TEST
          037140 104410          TRAP     C$ESCAPE
          037142 000754          .WORD   L10076
6858 037144 004537 004660    JSR      R5,INITT1    ;INITIALIZE TIMER # 1
6859 037150 000000          O           ; 0 ==> LATCHES
6860 037152 000300          BIT7+BIT6      ; MODE 3 & "T1" INT. ENABLE FLAG CLEARED
6861 037154 103003          BCC      .+10        ;IF NO ERROR, PROCEED
6862 037156          ERROR          ;ELSE, REPORT IT
          037156 104460          TRAP     C$ERROR
6863 037160          ESCAPE   TST           ; AND EXIT THIS TEST
          037160 104410          TRAP     C$ESCAPE
          037162 000734          .WORD   L10076
6864 037164 004737 036146    JSR      PC,GETT1     ;IS "T1" SET?
6865 037170 102002          BVC      .+6         ;IF NO ERROR, PROCEED
6866 037172          ESCAPE   SUB           ;ELSE, IT'S ALREADY BEEN REPORTED
          037172 104410          TRAP     C$ESCAPE
          037174 000720          .WORD   L10100
6867 037176 103006          BCC      2$           ;NO, GOOD.

```

TEST 28 VIA TIMER 1 FREE RUNNING MODE TEST

```

6868 037200          GEDF  EM50A,ERR50  ;YES, REPORT IT'S NOT BEING CLEARED @ INIT.
;          "DEVICE FATAL" ERROR # 92
          037200 104455          TRAP  C$ERDF
          037202 000134          .WORD 92
          037204 016067          .WORD EM50A
          037206 010762          .WORD ERR50
6869 037210          ESCAPE SUB          ; & EXIT TEST
          037210 104410          TRAP  C$ESCAPE
          037212 000702          .WORD L10100-.

6870
6871 ;-----
6872
6873 037214 004537 036114 2$: JSR  R5,LODT1C  ;RELOAD TIMER 1'S COUNTERS WITH NEW VALUES:
6874 037220 125 125 .BYTE 125,125
6875
6876 ;-----
6877
6878 037222 005003          CLR  R3          ;INITIALIZE TIMEOUT COUNTER
6879 037224 004737 036146 4$: JSR  PC,GETT1  ;"T1" SHOULD BE SET. IS IT?
6880 037230 102002          BVC  .+6        ;IF NO ERROR, PROCEED
6881 037232          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
          037232 104410          TRAP  C$ESCAPE
          037234 000660          .WORD L10100 .
6882 037236 103407          BCS  5$        ;YES, GOOD.
6883 037240 077307          SOB  R3,4$     ;NO, IF NO TIMEOUT, LOOK AGAIN
6884 037242          GEDF  EM50F,ERR50 ;ELSE, SAY IT WASN'T SET BY T1 TIMEOUT
;          "DEVICE FATAL" ERROR # 93
          037242 104455          TRAP  C$ERDF
          037244 000135          .WORD 93
          037246 016341          .WORD EM50F
          037250 010762          .WORD ERR50
6885 037252          ESCAPE SUB          ;IF ERROR, THE REST OF THIS TEST IS UN-DOABLE!
          037252 104410          TRAP  C$ESCAPE
          037254 000640          .WORD L10100-.

6886
6887 ;-----
6888
6889 037256          5$: JSR  PC,GETPB7  ;GET "PB7". IS IT SET?
6890 037256 004737 036332  BVC  .+6        ;IF NO ERROR, PROCEED
6891 037262 102002          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
          037264 104410          TRAP  C$ESCAPE
          037266 000626          .WORD L10100 .
6893 037270 103406          BCS  36$     ;YES, GOOD.
6894 037272          GEDF  EM50U,ERR50 ;NO, REPORT IT NOT SET.
;          "DEVICE FATAL" ERROR # 94
          037272 104455          TRAP  C$ERDF
          037274 000136          .WORD 94
          037276 017174          .WORD EM50U
          037300 010762          .WORD ERR50
6895 037302          ESCAPE SUB          ; & ALLOW RESTART OF THIS SUBTEST
          037302 104410          TRAP  C$ESCAPE
          037304 000610          .WORD L10100 .

6896
6897 ;-----
6898
6899 037306 004537 004064 36$: JSR  R5,READ  ;READING T1CH SHOULDN'T CLEAR T1

```


TEST 28 - VIA TIMER 1 FREE-RUNNING MODE TEST

```

6900 037312 120005          T1CH
6901 037314 002452          TMP5          ; (WE DON'T CARE WHAT THIS IS)
6902 037316 103003          BCC          .+10      ;IF NO ERROR, PROCEED
6903 037320. 104460          ERROR          ;ELSE, REPORT IT
                                TRAP      C$ERROR
6904 037322          ESCAPE TST          ;          AND EXIT THIS TEST
                                TRAP      C$ESCAPE
                                .WORD      L10076 .
6905 037326 004737 036146    JSR          PC,GETT1    ;CHECK "T1" -- IT SHOULD STILL BE SET
6906 037332 102002          BVC          .+6        ;IF NO ERROR, PROCEED
6907 037334          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
                                TRAP      C$ESCAPE
                                .WORD      L10100 .
6908 037340 103406          BCS          37$        ;IT IS, GOOD.
6909 037342          GEDF          EM50G,ERR50 ;CLEARED BY READING T1CH!!
                                ;          "DEVICE FATAL" ERROR # 95
                                TRAP      C$ERDF
                                .WORD      95
                                .WORD      EM50G
                                .WORD      ERR50
6910 037342 104455          ESCAPE SUB          ;          ALLOW RESTART OF THIS SUBTEST
                                TRAP      C$ESCAPE
                                .WORD      L10100-.
6911
6912          ; -----
6913
6914 037356 005003 37$: CLR          R3          ;INITIALIZE TIMEOUT COUNTER AGAIN
6915 037360 004737 036146    38$: JSR          PC,GETT1 ;WAIT FOR "T1" TO BE SET AGAIN
6916 037364 102002          BVC          .+6        ;IF NO ERROR, PROCEED
6917 037366          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
                                TRAP      C$ESCAPE
                                .WORD      L10100-.
6918 037372 103407          BCS          39$        ;GOT IT - NO CHECK PB7
6919 037374 077307          SOB          R3,38$    ;NOT YET. IF NO TIMEOUT, LOOK AGAIN.
6920 037376          GEDF          EM50L,ERR50 ;ELSE, TIMER NOT REALLY WORKING RIGHT!
                                ;          "DEVICE FATAL" ERROR # 96
                                TRAP      C$ERDF
                                .WORD      96
                                .WORD      EM50L
                                .WORD      ERR50
6921 037376 104455          ESCAPE SUB          ;
                                TRAP      C$ESCAPE
                                .WORD      L10100 .
6922
6923          ; -----
6924
6925 037412          39$: JSR          PC,GETPB7 ;GET "PB7". IS IT SET?
6926 037412 004737 036332    BVC          .+6        ;IF NO ERROR, PROCEED
6927 037416 102002          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT
                                TRAP      C$ESCAPE
                                .WORD      L10100-.
6928 037420 104410          BCS          6$         ;YES, GOOD.
6929 037422 000472          GEDF          EM50Z,ERR50 ;NO, REPORT "PB7" NOT SET AFTER SECOND CYCLE
6930 037424 103404          ;          "DEVICE FATAL" ERROR # 97
                                TRAP      C$ERDF
                                .WORD      97
6931 037426 104455
6932 037430 000141

```

TEST 28 - VIA TIMER 1 FREE-RUNNING MODE TEST

```

037432 017471
037434 010762
6931
6932
6933
6934 037436 004737 036204 6$: JSR PC,KICKT1 ;KICK IT OFF AGAIN SO WE CAN PRESERVE TIMING
6935 037442 103003 BCC .+10 ;IF NO ERROR, PROCEED
6936 037444 ERROR ;ELSE, REPORT IT
037444 104460 TRAP C$ERROR
6937 037446 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ESCAPE
037446 104410 TRAP C$ESCAPE
037450 000446 .WORD L10076-.

6938
6939
6940 037452 005003 ;WAIT FOR IT TO FINISH:
6941 037454 004737 036146 7$: CLR R3 ;INITIALIZE TIMEOUT COUNTER
6942 037460 102002 JSR PC,GETT1 ;"T1" SHOULD BE SET. IS IT?
6943 037462 BVC .+6 ;IF NO ERROR, PROCEED
037462 104410 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED - EXIT TRAP C$ESCAPE
037464 000430 .WORD L10100-.

6944 037466 103402 BCS 8$ ;YES, GOOD.
6945 037470 077307 SOB R3,7$ ;NO. IF NO TIMEOUT, LOOK AGAIN
6946 037472 000442 BR 14$ ;IF TIMEOUT, BYPASS NEXT CHECK (THIS DONE ABOVE)
6947 037474 004537 004064 8$: JSR R5,READ ;READING T1CL SHOULD CLEAR "T1"
6948 037500 120004 T1CL
6949 037502 002450 TMP4 ; (WE DON'T CARE WHAT THIS IS EITHER)
6950 037504 103003 BCC .+10 ;IF NO ERROR, PROCEED
6951 037506 ERROR ;ELSE, REPORT IT TRAP C$ERROR
037506 104460 ESCAPE TST ; AND EXIT THIS TEST TRAP C$ESCAPE
037510 104410 TRAP C$ESCAPE
037512 000404 .WORD L10076-.

6953 037514 004737 036146 JSR PC,GETT1 ;CHECK "T1" - IT SHOULD BE CLEARED NOW
6954 037520 102002 BVC .+6 ;IF NO ERROR, PROCEED
6955 037522 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT TRAP C$ESCAPE
037522 104410 TRAP C$ESCAPE
037524 000370 .WORD L10100-.

6956 037526 103006 BCC 9$ ;IT IS, GOOD.
6957 037530 GEDF EM50C,ERR50 ;NOT CLEARED! REPORT IT.
; "DEVICE FATAL" ERROR # 98 TRAP C$ERDF
037530 104455 TRAP C$ERDF
037532 000142 .WORD 98
037534 016203 .WORD EM50C
037536 010762 .WORD ERR50
6958 037540 ESCAPE SUB ;IF THIS ERROR OCCURED, EXIT SUBTEST TRAP C$ESCAPE
037540 104410 TRAP C$ESCAPE
037542 000352 .WORD L10100 .

6959 037544 005003 9$: CLR R3 ;RE-INITIALIZE THE TIMEOUT COUNTER
6960 037546 004737 036146 12$: JSR PC,GETT1 ;WAIT FOR "T1" TO GET SET AGAIN
6961 037552 102002 BVC .+6 ;IF NO ERROR, PROCEED
6962 037554 ESCAPE SUB ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT TRAP C$ESCAPE
037554 104410 TRAP C$ESCAPE
037556 000336 .WORD L10100-.

6963 037560 103407 BCS 14$ ;GOT IT - GOOD.
6964 037562 077307 SOB R3,12$ ;NOT YET. IF NO TIMEOUT, TRY AGAIN.
6965 037564 GEDF EM50X,ERR50 ;ELSE, REPORT "T1" NOT RESET
; "DEVICE FATAL" ERROR # 99

```

TEST 28 - VIA TIMER 1 FREE RUNNING MODE TEST

037564	104455								TRAP	C\$ERDF
037566	000143								.WORD	99
037570	017354								.WORD	EM50X
037572	010762								.WORD	ERR50
6966	037574				ESCAPE	SUB		;IF ERROR, CAN'T CONTINUE THIS TEST		
	037574	104410							TRAP	C\$ESCAPE
	037576	000316							.WORD	L10100--
6967	037600	112737	000252	002455	14\$:	MOVW	#252,TMP6+1	;SETUP FOR AND		
6968	037606	004537	004310			JSR	R5,WRITE	; LOAD T1LL (ADDR 6)		
6969	037612	120006				T1LL				
6970	037614	002455				TMP6+1		; WITH 252 OCTAL		
6971	037616	103003				BCC	.+10	;IF NO ERROR, PROCEED		
6972	037620					ERROR		;ELSE, REPORT IT		
	037620	104460							TRAP	C\$ERROR
6973	037622				ESCAPE	TST		; AND EXIT THIS TEST		
	037622	104410							TRAP	C\$ESCAPE
	037624	000272							.WORD	L10076--
6974	037626	004737	036146			JSR	PC,GETT1	;THIS SHOULDN'T CLEAR "T1"		
6975	037632	102002				BVC	.+6	;IF NO ERROR, PROCEED		
6976	037634					ESCAPE	SUB	;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT		
	037634	104410							TRAP	C\$ESCAPE
	037636	000256							.WORD	L10100--
6977	037640	103406				BCS	16\$;IT DIDN'T -- GOOD.		
6978	037642					GEDF	EM50M,ERR50	;WOOPS, IT DID!! REPORT FAILURE		
								; "DEVICE FATAL" ERROR # 100		
	037642	104455							TRAP	C\$ERDF
	037644	000144							.WORD	100
	037646	017012							.WORD	EM50M
	037650	010762							.WORD	ERR50
6979	037652				ESCAPE	SUB		; THE REST OF THIS TEST IS INVALID TOO!		
	037652	104410							TRAP	C\$ESCAPE
	037654	000240							.WORD	L10100--
6980	037656	004737	036332			JSR	PC,GETPB7	;"PB7" SHOULD BE LOW HERE		
6981	037662	102002				BVC	.+6	;IF NO ERROR, PROCEED		
6982	037664					ESCAPE	SUB	;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT		
	037664	104410							TRAP	C\$ESCAPE
	037666	000226							.WORD	L10100--
6983	037670	103054				BCC	17\$;IT WASN'T, GOOD.		
6984	037672	004537	004064			JSR	R5,READ	;GET IFR FOR ERROR MESSAGE		
6985	037676	120015				IFR				
6986	037700	002472				TMPD				
6987	037702	103003				BCC	.+10	;IF NO ERROR, PROCEED		
6988	037704					ERROR		;ELSE, REPORT IT		
	037704	104460							TRAP	C\$ERROR
6989	037706				ESCAPE	TST		; AND EXIT THIS TEST		
	037706	104410							TRAP	C\$ESCAPE
	037710	000206							.WORD	L10076 .
6990	037712	004537	004064			JSR	R5,READ	;GET T1CL FOR ERROR MESSAGE		
6991	037716	120004				T1CL				
6992	037720	002450				TMP4				
6993	037722	103003				BCC	.+10	;IF NO ERROR, PROCEED		
6994	037724					ERROR		;ELSE, REPORT IT		
	037724	104460							TRAP	C\$ERROR
6995	037726				ESCAPE	TST		; AND EXIT THIS TEST		
	037726	104410							TRAP	C\$ESCAPE
	037730	000166							.WORD	L10076 .
6996	037732	004537	004064			JSR	R5,READ	;GET T1CH FOR ERROR MESSAGE		

TEST 28 VIA TIMER 1 FREE-RUNNING MODE TEST

```

6997 037736 120005          T1CH
6998 037740 002452          TMP5
6999 037742 103003          BCC      .+10          ;IF NO ERROR, PROCEED
7000 037744          ERROR          ;ELSE, REPORT IT
7001 037746          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
      037746 104460          ;
      037750 000146          TRAP      C$ESCAPE
7002 037752 004537 004064          JSR      R5,READ          ;GET T1LL FOR ERROR MESSAGE          .WORD    L10076-.
7003 037756 120006          T1LL
7004 037760 002454          TMP6
7005 037762 103003          BCC      .+10          ;IF NO ERROR, PROCEED
7006 037764          ERROR          ;ELSE, REPORT IT
7007 037766          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
      037766 104410          ;
      037770 000126          TRAP      C$ESCAPE
7008 037772 004537 004064          JSR      R5,READ          ;GET T1LH FOR ERROR MESSAGE          .WORD    L10076-.
7009 037776 120007          T1LH
7010 040000 002456          TMP7
7011 040002 103003          BCC      .+10          ;IF NO ERROR, PROCEED
7012 040004          ERROR          ;ELSE, REPORT IT
7013 040006          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
      040006 104410          ;
      040010 000106          TRAP      C$ESCAPE
7014 040012          GEDF    EM50Y,ERR50          ;IT WAS! REPORT IT BEING RESET BY WRITTING T1LL          .WORD    L10076-.
      040012 104455          ;          "DEVICE FATAL" ERROR # 101
      040014 000145          TRAP      C$ERDF
      040016 017417          .WORD    101
      040020 010762          .WORD    EM50Y
7015          .WORD    ERR50
7016
7017          ; AT THE ABOVE "PB7" TEST, IT SHOULD BE LOW. NOT BECAUSE OF ANY READ/WRITE
7018          ; OPERATION, BUT BECAUSE OF WHERE WE ARE IN THE CYCLING OF TIMER # 1. "PB7"
7019          ; SHOULD BE LOW HERE UNTIL T1 TIMES OUT.
7020 040022 112737 000252 002457 17$: MOVB    #252,TMP7+1          ;SETUP FOR AND
7021 040030 004537 004310          JSR      R5,WRITE          ; LOAD T1LH (ADDR 7)
7022 040034 120007          T1LH
7023 040036 002457          TMP7+1          ; WITH 252 OCTAL
7024 040040 103003          BCC      .+10          ;IF NO ERROR, PROCEED
7025 040042          ERROR          ;ELSE, REPORT IT
7026 040044          ESCAPE TST          ;          AND EXIT THIS TEST          TRAP      C$ERROR
      040044 104410          ;
      040046 000050          TRAP      C$ESCAPE
7027 040050 004737 036146          JSR      PC,GETT1          ;THIS SHOULD CLEAR "T1"          .WORD    L10076 .
7028 040054 102002          BVC     .+6          ;IF NO ERROR, PROCEED
7029 040056          ESCAPE SUB          ;ELSE, IT'S ALREADY BEEN REPORTED -- EXIT
      040056 104410          TRAP      C$ESCAPE
      040060 000034          .WORD    L10100-.
7030 040062 103006          BCC     18$
7031 040064          GEDF    EM50A,ERR50          ;IT DID -- GOOD.
      040064 104455          ;NOP! REPORT: "T1" NOT CLEARED BY LOADING T1LH
      040066 000146          ;          "DEVICE FATAL" ERROR # 102
          TRAP      C$ERDF
          .WORD    102

```


HARDWARE PARAMETER CODING SECTION

.SBTTL HARDWARE PARAMETER CODING SECTION

7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070

```

; ////////////////////////////////////////////////////////////////////
; / THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
; / THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
; / MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
; / INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
; / MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
; / WITH THE OPERATOR.
; ////////////////////////////////////////////////////////////////////

```

```

040120          BGNHRD
040120 000015
040122
                                .WORD L10101 L$HARD/2
                                L$HARD::
7058 040122          GPRMA  ADDRES,0,0,160020,177776,YES
7059 040122 000031
                                .WORD T$CODE
040124 040154
                                .WORD ADDRES
040126 160020
                                .WORD T$LLOLIM
040130 177776
                                .WORD T$HILIM
7060 040132          GPRMA  VECTOR,2,0,0,674,YES
                                .WORD T$CODE
040132 001031
                                .WORD VECTOR
040134 040202
                                .WORD T$LLOLIM
040136 000000
                                .WORD T$HILIM
040140 000674
7061 040142          GPRMD  PRIRTY,4,0,7000,0,7,YES
                                .WORD T$CODE
040142 002032
                                .WORD PRIRTY
040144 040233
                                .WORD 7000
040146 007000
                                .WORD T$LLOLIM
040150 000000
                                .WORD T$HILIM
040152 000007
7062 040154          ENDRD
                                .EVEN
7063 040154
                                L10101:
7064 040154
7065 .NLIST BEX
7066 040154 104 105 126 ADDRES: .ASCIZ /DEVICE CSR ADDRESS : /
7067 040202 104 105 126 VECTOR: .ASCIZ /DEVICE VECTOR ADDRESS : /
7068 040233 104 105 126 PRIRTY: .ASCIZ /DEVICE PRIORITY LEVEL : /
7069 .LIST BEX
7070 .EVEN

```

SOFTWARE PARAMETER CODING SECTION

.SBTTL SOFTWARE PARAMETER CODING SECTION

7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084

```

;//////////////////////////////////////////////////////////////////
;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
;/ WITH THE OPERATOR.
;//////////////////////////////////////////////////////////////////

```

7085 040264
040264 000000
040266
7086 040266
040266

BGNSFT

ENDSFT

.WORD L10102 L\$SOFT/2
L\$SOFT::
.EVEN
L10102:

PATCH AREA FOR DEBUG

```

7088
7089 040266
7090          040366
7091 040366 000240
7092 040370 000240
7093 040372 000240
7094
7095
7096
7097
7098 040374
7099 040374

      040374 000000
      040376 000000
      040400
7100          000001

```

.SBTTL PATCH AREA FOR DEBUG

PATCH:

```

      . = . + 100
      NOP
      NOP
      NOP

```

.SBTTL "ENDMOD" & "LASTAD"

```

      ENDMOD
      LASTAD

```

```

L$LAST::
.END

```

```

.EVEN
.WORD 0
.WORD 0

```


SYMBOL TABLE

ACR = 120013	BSLT5 = 000025	C\$GPRI = 000040	EM50A 016067	FMT06A 012571
ADDRES 040154	BSLT6 = 000026	C\$JNIT = 000011	EM50B 016135	FMT06B 012637
ADR = 000020 G	BSLT7 = 000027	C\$INLP = 000020	EM50C 016203	FMT07 012530
AD.HIT 020350	BSR0 002246	C\$MANI = 000050	EM50D 016251	FMT10 012654
AD.OK 020344	BSR1 002250	C\$MEM = 000031	EM50E 016305	FMT11 012730
ASSEMB = 000010	BSR10 002264	C\$MSG = 000023	EM50F 016341	FMT4 012245
BDATA 002312	BSR11 002270	C\$OPEN = 000034	EM50G 016406	FMT4A 012305
BIT0 = 000001 G	BSR12 002272	C\$PNTB = 000014	EM50H 016450	FMT4B 012340
BIT00 = 000001 G	BSR13 002274	C\$PNTF = 000017	EM50I 016536	FMT4C 012345
BIT01 = 000002 G	BSR14 002276	C\$PNTS = 000016	EM50J 016600	FMT47A 007232
BIT02 = 000004 G	BSR15 002300	C\$PNTX = 000015	EM50K 016666	FMT47B 007263
BIT03 = 000010 G	BSR16 002302	C\$QIO = 000377	EM50L 016730	FMT47C 007322
BIT04 = 000020 G	BSR17 002304	C\$RDBU = 000007	EM50M 017012	FMT47E 007350
BIT05 = 000040 G	BSR2 002252	C\$REFG = 000047	EM50N 017054	FMT47G 007407
BIT06 = 000100 G	BSR3 002254	C\$RESE = 000033	EM50S 017122	FMT48A 010300
BIT07 = 000200 G	BSR4 002256	C\$REVI = 000003	EM50U 017174	FMT48B 010347
BIT08 = 000400 G	BSR5 002260	C\$RFLA = 000021	EM50V 017240	FMT48C 010402
BIT09 = 001000 G	BSR6 002262	C\$RPT = 000025	EM50W 017305	FMT48E 010437
BIT1 = 000002 G	BSR7 002264	C\$SEFG = 000046	EM50X 017354	FMT48F 010506
BIT10 = 002000 G	BT1 = 003122	C\$SPRI = 000041	EM50Y 017417	FMT48G 010551
BIT11 = 004000 G	BT2 = 003206	C\$SVEC = 000037	EM50Z 017471	FMT48H 010611
BIT12 = 010000 G	BUFARE 003122	C\$TPRI = 000013	EM6 014540	FMT48I 010644
BIT13 = 020000 G	CONSOL 002346	DDRA = 120003	EM7 014564	FMT5 012400
BIT14 = 040000 G	CONTIN 020122	DDR8 = 120002	EM8 014611	FMT5A 012443
BIT15 = 100000 G	CONTST 020222	DELAY1 002316	EM9 014634	FMT50A 012747
BIT2 = 000004 G	CSREGS = 000020	DELAY2 002320	ENDEMB 012114	FMT50B 013021
BIT3 = 000010 G	C\$AU = 000052	DEVMAP 002342	ENDT7 022642	FMT50C 013102
BIT4 = 000020 G	C\$AUTO = 000061	DEVPTR 002344	ERRBLK 002244 G	FMT50D 013142
BIT5 = 000040 G	C\$BRK = 000022	DFPTBL 002216 G	ERRFLG 002332	FMT50E 013157
BIT6 = 000100 G	C\$BSEG = 000004	DIAGMC = 000000	ERRMSG 002242 G	FMT50M 013174
BIT7 = 000200 G	C\$BSUB = 000002	DMVDAL = 000001	ERRNBR 002240 G	FRSPAS 002340
BIT8 = 000400 G	C\$CEFG = 000045	DMVPU = 000004	ERRTYP 002236 G	FRSTIM 002336
BIT9 = 001000 G	C\$CLCK = 000062	EF.CON = 000036 G	ERR1 005276 G	F\$AU = 000015
BOE = 000400 G	C\$CLEA = 000012	EF.NEW = 000035 G	ERR2 005304 G	F\$AUTO = 000020
BSEL 002352	C\$CLOS = 000035	EF.PWR = 000034 G	ERR3 005414 G	F\$BGN = 000040
BSELRS 003042	C\$CLP1 = 000006	EF.RES = 000037 G	ERR4 005426 G	F\$CLEA = 000007
BSELO 002352	C\$CVEC = 000036	EF.STA = 000040 G	ERR4\$ 011322	F\$DU = 000016
BSEL1 002354	C\$DCLN = 000044	EM14 014667	ERR47 006724 G	F\$END = 000041
BSEL10 002372	C\$DODU = 000051	EM15 014733	ERR47. 007110	F\$HARD = 000004
BSEL11 002374	C\$DRPT = 000024	EM16 015013	ERR48 007632 G	F\$HW = 000013
BSEL12 002376	C\$DU = 000053	EM17 015110	ERR48. 010030	F\$INIT = 000006
BSEL13 002400	C\$EDI = 000003	EM17A 015170	ERR5 005552 G	F\$JMP = 000050
BSEL14 002402	C\$ERDF = 000055	EM20 015243	ERR5\$ 011710	F\$MOD = 000000
BSEL15 002404	C\$ERHR = 000056	EM20A 015275	ERR50 010762 G	F\$MSG = 000011
BSEL16 002406	C\$ERRO = 000060	EM20B 015355	ERR6 005650 G	F\$PROT = 000021
BSEL17 002410	C\$ERSF = 000054	EM21 015442	ERR7 006612 G	F\$PWR = 000017
BSEL2 002356	C\$ERSO = 000057	EM22 015477	ER47CT 007104	F\$RPT = 000012
BSEL3 002360	C\$ESCA = 000010	EM22A 015532	ER47MX 007106	F\$SEG = 000003
BSEL4 002362	C\$ESEG = 000005	EM25 015565	ER48CT 010024	F\$SOFT = 000005
BSEL5 002364	C\$ESUB = 000003	EM3 014454	ER48MX 010026	F\$SRV = 000010
BSEL6 002366	C\$ETST = 000001	EM34 015613	EVL = 000004 G	F\$SUB = 000002
BSEL7 002370	C\$EXIT = 000032	EM34B 015644	EXECUT = 000005	F\$SW = 000014
BSLT0 = 000020	C\$GETB = 000026	EM4 014500	E\$END = 002100	F\$TEST = 000001
BSLT1 = 000021	C\$GETW = 000027	EM47A 015675	E\$LOAD = 000035	GDATA 002310
BSLT2 = 000022	C\$GMAN = 000043	EM47B 015735	FMT02 012124	GETBSR 004434
BSLT3 = 000023	C\$GPHR = 000042	EM48A 016016	FMT02A 012161	GETPB7 036332
BSLT4 = 000024	C\$GPL0 = 000030	EM5 014515	FMT06 012562	GETPRM 020012

SYMBOL TABLE

GETT1	036146	I\$PTAB=	000041	L\$PRIO	002042 G	L10054	027356	ORA	=	120001	
GETWSR	004576	I\$PWR	=	000041	L\$PROT	017614 G	L10055	027352	ORAM	=	120017
G\$CNTD=	000200	I\$RPT	=	000041	L\$PRT	002112 G	L10056	027440	ORB	=	120000
G\$DELM=	000372	I\$SEG	=	000041	L\$REPP	002062 G	L10057	027434	O\$APTS=	000000	
G\$DISP=	000003	I\$SETU=	000041	L\$REV	002010 G	L10060	027524	O\$AU	=	000001	
G\$EXCP=	000400	I\$SFT	=	000041	L\$SOFT	040266 G	L10061	027520	O\$BGNR=	000000	
G\$HILI=	000002	I\$SRV	=	000041	L\$SPC	002056 G	L10062	027612	O\$BGNS=	000000	
G\$LOLI=	000001	I\$SUB	=	000041	L\$SPCP	002020 G	L10063	027606	O\$DU	=	000001
G\$NO	=	I\$TST	=	000041	L\$SPTP	002024 G	L10064	030100	O\$ERRT=	000001	
G\$OFFS=	000400	J\$JMP	=	000167	L\$STA	002030 G	L10065	030230	O\$GNSW=	000000	
G\$OFFSI=	000376	KICKT1	036204	L\$SW	002236 G	L10066	030360	O\$POIN=	000001		
G\$PRMA=	000001	LODT1C	036114	L\$TEST	002114 G	L10067	030514	O\$SETU=	000000		
G\$PRMD=	000002	LOE	=	040000 G	L\$TIML	002014 G	L10070	030644	PATA	002504	
G\$PRML=	000000	LOGDEV	002322	L\$UNIT	002012 G	L10071	030774	PATB	002526		
G\$RADA=	000140	LOT	=	000010 G	L10000	002234	L10072	031132	PATC	002556	
G\$RADB=	000000	LSIDCL=	000002	L10001	002236	L10073	036112	PATCH	040266		
G\$RADD=	000000	LSIMLT=	000020	L10002	005220	L10074	033404	PATCM	002624		
G\$RADL=	000120	LUIMOD	002000 G	L10003	005272	L10075	036110	PATCR	002604		
G\$RADO=	000020	L\$ACP	002110 G	L10004	005302	L10076	040116	PATD	002644		
G\$XFER=	000004	L\$APT	002036 G	L10005	005412	L10077	037050	PATE	002674		
G\$YES	=	L\$AU	020400 G	L10006	005424	L10100	040114	PATF	002724		
HELP	=	L\$AUT	002070 G	L10007	005550	L10101	040154	PATG	003040		
HOE	=	L\$AUTO	020232 G	L10010	005646	L10102	040266	PATGEN	023776		
IBE	=	L\$CCP	002106 G	L10011	006610	MASCLR	003614	PCR	=	120014	
IDU	=	L\$CLEA	020356 G	L10012	006722	MCLR	=	000100	PFLAG	002350	
IENBA	=	L\$CO	002032 G	L10013	007102	MLWRI	004332	PNT	=	001000 G	
IENBB	=	L\$DEPO	002011 G	L10014	010022	MPCSR	002352	PRI	=	002000 G	
IENR	=	L\$DESC	003542 G	L10015	011274	MPIHAN	005152 G	PRIPTY	040233		
IER	=	L\$DESP	002076 G	L10017	020220	MPIVEC	002412	PRI00	=	000000 G	
IFR	=	L\$DEVP	002060 G	L10020	020346	MPOHAN	005224 G	PRI01	=	000040 G	
IFRCA1=	000002	L\$DISP	002124 G	L10021	020372	MPOVEC	002414	PRI02	=	000100 G	
IFRCA2=	000001	L\$DLY	002116 G	L10022	020376	MPRIOR	002416	PRI03	=	000140 G	
IFRCB1=	000020	L\$DTP	002040 G	L10023	020400	MRDY	=	000200	PRI04	=	000200 G
IFRCB2=	000010	L\$DTYP	002034 G	L10024	020536	MREQ	=	000001	PRI05	=	000240 G
IFRIRQ=	000200	L\$DU	020374 G	L10025	020666	MSTCLR	003762	PRI06	=	000300 G	
IFRSR	=	L\$DUT	002072 G	L10026	021254	NEWLIN	012121	PRI07	=	000340 G	
IFRT1	=	L\$DVTY	003522 G	L10027	021446	NEWPC	002500	PSTACK	002324		
IFRT2	=	L\$EF	002052 G	L10030	021610	NEWST	017772	PU24	=	000001	
IHILNK	005222	L\$ENVI	002044 G	L10031	021732	NPRABT=	000200	RCRAM	023616		
IHOLNK	005274	L\$ERRT	002236 G	L10032	022134	NPRAIH=	000075	READ	004064		
INITT1	004660	L\$ETP	002102 G	L10033	022644	NPRAIL=	000074	READI	004176		
INTFLG	002326	L\$EXP1	002046 G	L10034	022354	NPRAIX=	000076	REDLOC=	000001		
INTWCH	002330	L\$EXP4	002064 G	L10035	022642	NPRAOH=	000071	REDPAG=	000003		
IRQA	=	L\$EXP5	002066 G	L10036	023212	NPRAOL=	000070	REGNUM	002334		
IRQB	=	L\$HARD	040122 G	L10037	023400	NPRAOX=	000072	REG0	002420		
IRQREG=	123005	L\$HIME	002120 G	L10040	025140	NPRBS7=	000010	REG1	002422		
ISR	=	L\$HPCP	002016 G	L10041	025762	NPRBYT=	000010	REG2	002424		
IXE	=	L\$HPTP	002022 G	L10042	026044	NPRCTL=	123004	REG3	002426		
I\$AU	=	L\$HW	002216 G	L10043	026040	NPRDL	=	000044	REG4	002430	
I\$AUTO=	000041	L\$ICP	002104 G	L10044	026126	NPRDLB=	000054	REG5	002432		
I\$CLN	=	L\$INIT	017622 G	L10045	026122	NPRDRH=	123001	REG6	002434		
I\$DU	=	L\$LADP	002026 G	L10046	026230	NPRDRL=	123000	REG7	002436		
I\$HRD	=	L\$LAST	040400 G	L10047	026224	NPRGO	=	000100	RESFMC	003040	
I\$INIT=	000041	L\$LOAD	002100 G	L10050	027274	NPRI0	=	000040	RESFT3	003062	
I\$MOD	=	L\$LUN	002074 G	L10051	026736	NPRLD	=	000004	RESTR	017766	
I\$MSG	=	L\$MREV	002050 G	L10052	027114	NULERR	012072	RUN	=	000200	
I\$PRDT=	000040	L\$NAME	002000 G	L10053	027272	OLDSP	002502	SELO	002352		

SYMBOL TABLE

SEL10	002372	TXTVRB	014430	T\$LOLI=	000000	T1.2	021001	T4	021450	G
SEL12	002376	TXTVRC	014434	T\$LSYM=	010000	T1.3	021034	T5	021612	G
SEL14	002402	TXTVRD	014440	T\$LTNO=	000034	T1.4	021106	T6	021734	G
SEL16	002406	TXTVRE	014444	T\$NEST=	177777	T10	024162	T7	022136	G
SEL2	002356	TXTVRF	014450	T\$NS0 =	000000	T11	025142	T7.1	022136	
SEL4	002362	TXTVRT	017554	T\$NS1 =	000005	T12	025764	T7.2	022356	
SEL6	002366	TXTVR0	014345	T\$NS2 =	000002	T12.LP	026006	T8	022646	G
SFPTBL	002236	TXTVR1	014351	T\$NS3 =	000003	T12.1	026006	T9	023214	G
SLT0 =	000020	TXTVR2	014355	T\$PTNU=	000000	T13	026046	T9.RST	023402	
SLT2 =	000022	TXTVR3	014362	T\$SAVL=	177777	T13.LP	026070	UAM =	000200	G
SLT4 =	000024	TXTVR4	014367	T\$SEGL=	177777	T13.1	026070	VECTOR	040202	
SLT6 =	000026	TXTVR5	014374	T\$SEKO=	010000	T14	026130	WA =	003346	
SR =	120012	TXTVR6	014401	T\$SUBN=	000002	T14.LP	026172	WB =	003350	
STALL	005032	TXTVR7	014406	T\$TAGL=	177777	T14.1	026172	WC =	003352	
STARST	017722	TXTVR8	014413	T\$TAGN=	010103	T15	026232	WD =	003354	
STREG	005034	TXTVR9	014420	T\$TEMP=	000000	T15.1	026266	WE =	003356	
SVCGBL=	000000	TXT1	013253	T\$TEST=	000034	T15.2	026740	WF =	003360	
SVCINS=	000001	TXT2	013311	T\$TSTM=	177777	T15.3	027116	WRCRAM	023420	
SVCSUB=	000001	TXT2A	013353	T\$TSTS=	000001	T16	027276	WRILOC=	000000	
SVCTAG=	000001	TXT2B	013412	T\$\$AU =	010023	T16.LP	026276	WRIPAG=	000000	
SVCTST=	000001	TXT3	013454	T\$\$AUT=	010020	T16.L1	026750	WRITE	004310	
SWPBOT=	121000	TXT4	013504	T\$\$CLE=	010021	T16.L2	027126	WRITEI	004322	
SWPDDC=	121400	TXT4A	013544	T\$\$DU =	010022	T16.1	027320	WSR0	002246	
S\$LSYM=	010000	TXT47C	007442	T\$\$HAR=	010101	T17	027360	WSR10	002256	
TDATA	002306	TXT47D	007453	T\$\$HW =	010000	T17.1	027402	WSR12	002260	
TMPA	002464	TXT47E	007466	T\$\$INI=	010017	T18	027442	WSR14	002262	
TMPB	002466	TXT47F	007510	T\$\$MSG=	010025	T18.LP	027320	WSR16	002264	
TMPC	002470	TXT47G	007533	T\$\$PRO=	010016	T18.1	027466	WSR2	002250	
TMPD	002472	TXT47H	007556	T\$\$SEG=	010000	T19	027526	WSR4	002252	G
TMPE	002474	TXT47P	007614	T\$\$SOF=	010102	T19.LP	027402	WSR6	002254	
TMPF	002476	TXT48A	010736	T\$\$SRV=	010003	T19.1	027554	W0 =	003322	
TMP0	002440	TXT48B	010743	T\$\$SUB=	010100	T2	021146	W1 =	003324	
TMP1	002442	TXT48C	010750	T\$\$SW =	010001	T2CH =	120011	W2 =	003326	
TMP2	002444	TXT48D	010755	T\$\$TES=	010076	T2CL =	120010	W3 =	003330	
TMP3	002446	TXT5	013605	T.EDF =	000001	T2LL =	120010	W4 =	003332	
TMP4	002450	TXT6	013607	T.EHRD=	000002	T20	027614	W5 =	003334	G
TMP5	002452	TXT7	013632	T.ESF =	000000	T20.LP	027466	W6 =	003336	
TMP6	002454	TXT7A	013717	T.ESFT=	000003	T21	030102	W7 =	003340	G
TMP7	002456	TXT8A	014004	T1	020402	T21.LP	027554	W8 =	003342	
TMP8	002460	TXT8B	014021	T1CH =	120005	T22	030232	W9 =	003344	
TMP9	002462	TXT8C	014036	T1CL =	120004	T23	030362	XDATA	002314	
TXTMLT	017532	TXT8D	014053	T1LH =	120007	T24	030516	XORGB	011276	
TXTML0	014105	TXT8E	014070	T1LHGO=	120005	T25	030646	X\$ALWA=	000000	
TXTML1	014111	T\$ARGC=	000001	T1LL =	120006	T26	030776	X\$FALS=	000040	
TXTML2	014125	T\$CODE=	002032	T1.EHD	020670	T27	031134	X\$OFFS=	000400	G
TXTML3	014142	T\$ERRN=	000146	T1.EM1	020552	T27.1	031134	X\$TRUE=	000020	
TXTML4	014164	T\$EXCP=	000000	T1.HIT	020540	T27.2	033406	\$E =	000146	
TXTML5	014205	T\$FLAG=	000040	T1.HSB	020550	T28	036366	\$LSTIN=	000001	
TXTML6	014235	T\$GMAN=	000000	T1.HSW	020546	T28.1	036366	\$LSTTA=	000001	
TXTML7	014247	T\$HILI=	000007	T1.OK	020536	T28.2	037052	\$MPCSR=	160000	G
TXTVR	014327	T\$LAST=	000001	T1.1	020717	T3	021256	\$T =	000034	
TXTVRA	014425									

. ABS. 040400 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 31952 WORDS (125 PAGES)

SYMBOL TABLE

DYNAMIC MEMORY: 19748 WORDS (75 PAGES)
ELAPSED TIME: 00:08:46
CNDMAA.BIC,CNDMAA.SEQ/CR/ SP=SVC34.MLB/ML,CNDMAA.P11

.