

BDV11-AA

BOOTSTRAP DIAGNOSTIC
CVM8AB0

AH-B062B-MC
COPYRIGHT '77-78
FICHE 1 OF 1

JAN 1979
digital
MADE IN USA

This microfiche card contains a grid of frames, each displaying diagnostic data for the CVM8AB0 system. The data is organized into columns and rows, with some frames containing headers such as 'CPU', 'MEM', 'DISK', and 'I/O'. The frames show various numerical values and status indicators, likely representing system performance metrics or error codes. The data is presented in a structured, tabular format, typical of diagnostic reports from this era.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

002000

.ENABL ABS,AMA
.=2000
.NLIST CND,MD,MEB,ME
.TITLE USER DOCUMENTATION
.SBTTL IDENTIFICATION

: PRODUCT CODE: AC-B061B-MC

: PRODUCT NAME: CVM8AB0 BDV11-AA DIAG

: MAINTAINER: DIAGNOSTIC ENGINEERING

: AUTHOR: MARY MCNALLY 18-AUG-77

: REVISED: BILL HEAVEY NOV,1978

:
: COPYRIGHT (C) 1977,1978
: DIGITAL EQUIPMENT CORPORATION, MAYNARD MASSACHUSSETTS 01754

: THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
: SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLU-
: SION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY
: OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE
: AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM
: AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND
: OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

: THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
: NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
: EQUIPMENT CORPORATION.

: DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
: ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
:

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

+++
FUNCTIONAL DESCRIPTION: THE BDV11-AA BOOTSTRAP/TERMINATOR/
DIAGNOSTIC MODULE PROVIDES THE
FOLLOWING FUNCTIONS:
1. ROM RESIDENT HARDWARE DIAGNOSTIC TESTS.
2. PADS FOR ROM RESIDENT BOOTSTRAP ROUTINES FOR THOSE DEVICES WHICH ARE SUPPORTED BY THE LSI-11 SYSTEM.
3. A READ/WRITE STORAGE REGISTER FOR USE BY THE RESIDENT DIAGNOSTIC TESTS.
4. TWELVE DIP ROCKER SWITCHES TO SELECT TESTING AND BOOTSTRAP OPTIONS AT POWER UP.
5. AN ARRAY OF FOUR LED'S TO PROVIDE STATUS INFORMATION.
6. HALT AND REBOOT TOGGLE SWITCHES FOR USE IN SYSTEMS WITHOUT A CONSOLE.
7. SOCKETS FOR 2K WORDS OF EPROM.
8. OPTIONAL REPLACEMENT OF SYSTEM ROM BY 8K WORDS OF EPROM.

--
; VERSION: 00

81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133

.SBTTL GENERAL PROGRAM INFORMATION

: PROGRAM PURPOSE: THIS DIAGNOSTIC WILL BE USED TO ESTABLISH
: CONFIDENCE THAT THE MODULE IS FUNCTIONING
: PROPERLY. IT WILL PROVIDE CHECKSUM VERI-
: FICATION OF THE CONTENTS OF THE DIAGNOSTIC
: ROMS AND ANY ADDITIONAL ROM OR EPROM. IN
: ADDITION, IT WILL VERIFY THAT THE PROPER
: DIAGNOSTIC ROMS ARE INSERTED IN THE MODULE
: BY COMPARING THE ACTUAL CHECKWORDS IN THE
: ROMS TO THOSE SPECIFIED IN THE DIAGNOSTIC
: PROGRAM. IT WILL ALSO ACCEPT CHECKWORDS
: FROM AN OPERATOR FOR USE IN TESTING ANY
: ADDITIONAL ROM/EPROM. THE DIAGNOSTIC WILL
: ALSO TEST THE PROGRAMMABLE REGISTERS AND
: EXERCISE THE LED'S FOR OPERATOR INSPECTION.

: SYSTEM REQUIREMENTS:
: HARDWARE: LSI-11 PROCESSOR
: 16K WORDS OF MEMORY
: CONSOLE TERMINAL
: DIAGNOSTIC PROGRAM LOAD DEVICE

: RELATED DOCUMENTS AND STANDARDS:
: DIAGNOSTIC SUPERVISOR FUNCTIONAL SPEC (176-681-001)
: APT/DIAGNOSTIC SUPERVISOR INTERFACE SPEC (176-681-003)

: DIAGNOSTIC HIERARCHY PREREQUISITES: NONE, ALTHOUGH IT IS ASSSUMED THAT
: THE CPU IS FUNCTIONING PROPERLY.

: ASSUMPTIONS:
: --WHEN RUNNING UNDER APT, ALL ROCKER
: SWITCHES ARE IN THE 'ON' POSITION.
: THE EXCEPTION TO THIS OCCURS ONLY
: WHEN AN OPERATOR CHANGES THE HARD-
: WARE P-TABLE TO CORRESPOND TO THE NEW
: SWITCH SETTINGS.
: --THE ADDRESS JUMPERS ARE CONFIGURED
: AND MEMORY CHIPS INSTALLED PROPERLY.
: NO TWO CHIPS CAN RESPOND TO THE SAME
: ADDRESS.
: --THE MODULE UNDER TEST RESIDES IN THE
: SAME BACKPLANE AS THAT FROM WHICH THE
: LINE TIME CLOCK IS GENERATED.
: --THE CPU IS WORKING PROPERLY.

134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188

.SBTTL OPERATING INSTRUCTIONS

:1. LOADING AND STARTING PROCEDURES

: IN SYSTEMS OTHER THAN APT, BOTH THE DIAGNOSTIC PROGRAM
: AND THE DIAGNOSTIC SUPERVISOR WILL BE LOADED BY EITHER PAPER
: TAPE OR XXDP MEDIA. THE COMBINED FILE WILL BE CALLED CVMBAB,
: AND IS LOADED BY THE FOLLOWING COMMANDS:

I. PAPER TAPE

TO LOAD, PLACE AN ABSOLUTE LOADER IN THE PAPER
TAPE READER, AND TYPE '177550L'. THEN PLACE THE
CVMBAB TAPE IN THE READER AND TYPE 'P'.

TO CALL THE SUPERVISOR, TYPE '200G'. THE SUPERVISOR
WILL RESPOND WITH A FEW QUESTIONS AND A PROMPT CHARACTER.
SEE THE SUPERVISOR COMMANDS BELOW FOR FURTHER INSTRUCTIONS.

II. XXDP MEDIA

TO LOAD, TYPE 'L CVMBAB'. TO CALL THE
SUPERVISOR, TYPE 'S 200' WHEN THE PROGRAM IS LOADED.

III. SUPERVISOR COMMANDS

ONCE THE SUPERVISOR HAS BEEN INVOKED AT LOCATION 200,
THE FOLLOWING COMMANDS SHOULD BE USED SELECTIVELY TO
CONTROL THE RUNNING OF THE DIAGNOSTIC:

:2. TO START

START/TEST:<TESTNOS>/PASS:<PASSCNT>/UNIT:<DEVN>/FLAG:<CF>:<CF>

WHERE:

TEST ::= (DEFINES WHICH TESTS TO EXECUTE, IF NO
SPECIFICATION EXECUTE ALL TESTS)
PASS ::= (INDICATES HOW MANY PASSES TO RUN, IF NO SPEC-
IFICATION RUN UNTIL DIAGNOSTIC ESCAPE SEQUENCE)
UNIT ::= (SPECIFIES WHICH UNIT ENTRIES TO GET FROM THE
CONFIGURATION FILE, IF NO SPECIFICATION USE ALL
APPLICABLE UNIT ENTRIES)
FLAG ::= (SPECIFIES THE ERROR CONTROL/REPORT FLAG OPTIONS
TO BE USED)
<TESTNOS> ::= (LIST FOR UP TO 16 TESTS TO BE EXECUTED IN AN
ASCENDING ORDER.)
<PASSCNT> ::= (NUMBER OF PROGRAM PASSES TO EXECUTE)
<DEVN> ::= (UNIQUE, DEC STANDARD, DEVICE SPECIFIER AND
UNIT NUMBER)
<CF> ::= (ANY OF THE FOLLOWING CONTROL FLAGS:
HOE-HALT ON ERROR
LOE-LOOP ON ERROR AND ATTEMPT REPORT
IER-INHIBIT ALL ERROR REPORTS
IBE-INHIBIT BASIC AND EXTENDED ERROR REPORTS
IEE-INHIBIT EXTENDED ERROR REPORTS
PRI-DIRECT ALL ERROR, PASS, AND STATISTICAL
REPORTS TO THE LINE PRINTER.
BOE-AUDIO ERROR INDICATION
UAM-UNATTENDED MODE, NO OPERATOR INTERVENTION
PNT-PRINT NUMBER OF TEST BEING EXECUTED.)

189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225

:3. TO RESTART
: THE RESTART COMMAND IS SIMILAR TO THE START COMMAND EXCEPT
: THAT ALL PARAMETERS ARE ASSUMED TO BE ALREADY DEFINED, AND NO
: OPERATOR DIALOGUE IS PERFORMED PRIOR TO RUNNING THE DIAGNOSTIC.
: IF THE OPERATOR WISHES TO ALTER THE TYPE OF ADDITIONAL MEMORY
: TO TEST, OR CHANGE THE ADDRESSES, LOCATION 'PASS' MUST BE
: CLEARED MANUALLY PRIOR TO RESTARTING, SINCE THIS INFORMATION
: IS SET UP ON THE FIRST PASS OF THE DIAGNOSTIC.
:
: RESTART/TEST:<TESTNOS>/PASS:<PASSCNT>/FLAG:<CF>:<CF>...
:4. TO RETURN TO PROGRAM
:
: TO RESUME EXECUTION OF THE DIAGNOSTIC AT THE FIRST INSTRUCTION
: FOLLOWING THE CURRENT SUPERVISOR CALL, AT WHICH TIME NEW FLAGS
: MAY BE ASSIGNED.
:
: CONTINUE/FLAG:<CF>:<CF>:....
:5. TO LOAD AND START THE DIAGNOSTIC
: TO LOAD AND START THE DIAGNOSTIC USING DEFAULT PARAMETERS
:
: RUN<FILESPEC>/TEST:<TESTNOS>/PASS:<PASSCNT>/UNIT:<DEVN>/FLAG:<CF>...
:6. TO RETURN TO SUPERVISOR
:
: EXIT
:
: NOTE: TEST NUMBERS AND UNIT NUMBERS MAY BE SPECIFIED
: AS SINGLE NUMBERS, RANGES OF NUMBERS (I.E. 1-6),
: OR COMBINATIONS OF BOTH.
:
: SPECIAL ENVIRONMENTS: APT
: TEST 7, THE TEST OF ALL RESIDENT MEMORY, WILL NOT RUN
: UNDER APT, AS IT REQUIRES USER INTERVENTION.

226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265

: PROGRAM OPTIONS:
: THE HARDWARE PARAMETERS ARE STORED IN A PARAMETER TABLE WITH
: DEFAULT VALUES. THE OPERATOR WILL HAVE THE OPTION OF CHANGING
: THESE PARAMETERS BY RESPONDING TO THE APPROPRIATE QUESTIONS
: GENERATED BY THE DIAGNOSTIC SUPERVISOR. THESE PARAMETERS
: INCLUDE THE UNIT NUMBER, INTERRUPT VECTOR, PRIORITY LEVEL, AND
: ROCKER SWITCH SETTINGS. THE DEFAULT VALUES WILL BE TYPED ALONG
: WITH THE QUESTIONS.
: THE ROCKER SWITCH SETTINGS ARE EXAMINED IN THE FOLLOWING
: ORDER:
: B4 B3 B2 B1 A8 A7 A6 A5 A4 A3 A2 A1
: FOR EXAMPLE, IF SWITCHES A1, A2, A6, AND B1 WERE ON, THE SWITCH
: SETTING WOULD BE:
: B4 B3 B2 B1 A8 A7 A6 A5 A4 A3 A2 A1
: 1 1 1 1
: WHICH HAS AN OCTAL VALUE OF 0443.
: THE SOFTWARE P-TABLE CONTAINS THE CHECKWORDS FOR THE 2K
: OF DIAGNOSTIC ROM WHICH IS RESIDENT ON THE BDV11A. TO CHANGE
: THESE CHECKWORDS, THE OPERATOR MUST RESPOND WITH A YES TO THE
: SUPERVISOR'S QUESTION 'CHANGE SW (Y/N)?'. THE DEFAULT VALUES WILL
: THEN BE PRINTED AS THE QUESTIONS ARE ASKED.
:
: TEST 7 CHECKS ALL THE ADDITIONAL MEMORY THAT IT IS
: INSTRUCTED TO TEST. THIS TEST IS SET UP BY THE OPERATOR ON THE
: FIRST PASS OF THE DIAGNOSTIC. THE DIAGNOSTIC WILL ASK IF THERE
: IS ANY ADDITIONAL MEMORY TO TEST, AND IF SO WILL ASK WHICH
: TYPE OF MEMORY IT IS. (THE OPERATOR CAN ANSWER THESE QUESTIONS
: WITH LOGICAL Y/N ANSWERS.) IF ANY ADDITIONAL MEMORY IS TO BE
: TESTED, THE OPERATOR MUST SUPPLY THE CHECKWORDS FOR THOSE
: ROMS/EPROMS. IN THE CASE OF SYSTEM ROM/EPROM, THE OPERATOR WILL
: ALSO HAVE TO INDICATE HOW MANY CHECKWORDS WILL BE INPUT (IN DECIMAL).
: NOTE THAT ONCE THIS DATA IS SET UP, THIS MEMORY WILL ALWAYS BE
: TESTED, EVEN IF THE DIAGNOSTIC IS RESTARTED, UNLESS THE LOCATION
: 'PASS' IS CLEARED (SEE SEC.3 OF LOADING AND STARTING PROCEDURES).
:
: EXECUTION TIMES: A SINGLE ERROR-FREE PASS WILL REQUIRE
: LESS THAN 1 SEC. TO RUN UNDER APT. WHEN RUN
: IN STAND-ALONE MODE, IT WILL REQUIRE LESS
: THAN 3 SECS. TO RUN.

266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285

.SBTTL ERROR INFORMATION
: ERROR REPORTING PROCEDURES:
: IN GENERAL, ALL ERROR REPORTS WILL CONTAIN THE FOLLOWING
: INFORMATION:
: 1. A HEADER OF TEST IDENTIFICATION INFORMATION.
: THIS INCLUDES THE PROGRAM NAME, TYPE OF ERROR,
: ERROR NUMBER, TEST AND SUBTEST NUMBERS, UNIT
: NUMBER, AND AN OPTIONAL ADDITIONAL MESSAGE.
: 2. BASIC ERROR INFORMATION.
: THIS IS A SPECIFIC STATEMENT OF WHAT THE ERROR
: IS AND WHICH REGISTER OR ROM WAS INVOLVED.
: 3. EXTENDED ERROR INFORMATION.
: THIS IS OPTIONAL INFORMATION WHICH IS USED
: PRIMARILY TO GIVE THE EXPECTED AND ACTUAL
: CONTENTS OF THE APPROPRIATE DEVICE REGISTER
: DURING REGISTER TESTS.

	SBTTL	SUBTEST SUMMARIES			
			TEST NO.	SUBTEST NO.	PURPOSE
286		1	1	TO VERIFY THAT THE READ/WRITE REGISTER CAN HOLD ALL ZEROES.
287			2	TO VERIFY THAT THE READ/WRITE REGISTER CAN HOLD ALL ONES.
288			3	TO VERIFY THAT THE READ/WRITE REGISTER CAN HOLD AN ALTERNATING 1'S AND 0'S BIT PATTERN.
289			4	TO VERIFY THAT THE READ/WRITE REGISTER IS BYTE ADDRESSABLE.
290			5	TO VERIFY THAT THE READ/WRITE REGISTER CAN SWAP BYTES.
291			6	TO VERIFY THAT THE READ/WRITE REGISTER CAN HOLD AN ALTERNATING 0' AND 1'S BIT PATTERN.
292			7	TO VERIFY THAT THE READ/WRITE REGISTER IS BYTE ADDRESSABLE.
293			8	TO VERIFY THAT THE READ/WRITE REGISTER CAN SWAP BYTES.
294			9	TO VERIFY THAT THE READ/WRITE REGISTER CAN ROTATE A SET BIT WITHOUT PICKING UP ANY BITS.
295			10	TO VERIFY THAT THE READ/WRITE REGISTER CAN ROTATE A CLEAR BIT WITHOUT PICKING UP ANY BITS.
296		2	1	TEST 2 IS THE SAME AS TEST 1 EXCEPT THAT THE PAGE CONTROL REGISTER IS THE REGISTER UNDER TEST.
297			2	SAME AS TEST 1.
298			3	SAME AS TEST 1.
299			4	SAME AS TEST 1.
300			5	SAME AS TEST 1.
301			6	SAME AS TEST 1.
302			7	SAME AS TEST 1.
303			8	SAME AS TEST 1.
304			9	SAME AS TEST 1.
305			10	SAME AS TEST 1.
306		3	1	TO VERIFY THAT THE BEVENT CLAMP DISABLE ALLOWS INTERRUPTS WHEN OFF.
307			2	TO VERIFY THAT THE BEVENT CLAMP DISABLE INHIBITS INTERRUPTS WHEN ON.
308		6	1	TO VERIFY THAT THE LOW BYTE DIAGNOSTIC ROM HAS GOOD DATA.
309			2	TO VERIFY THAT THE HIGH BYTE DIAGNOSTIC ROM HAS GOOD DATA.
310			3	TO INSURE THAT THE DIAGNOSTIC ROMS HAVE NOT BEEN INTERCHANGED.
311		7	4	TO DETERMINE IF THERE IS ANY ADDITIONAL MEMORY TO TEST.
312			1	THIS INFORMATION IS OBTAINED
313				
314				
315				
316				
317				
318				
319				
320				
321				
322				
323				
324				
325				
326				
327				
328				
329				
330				
331				
332				
333				
334				
335				
336				
337				
338				
339				
340				
341				

342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357

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

2

3

4

5

THROUGH USER DIALOGUE.
TO TEST THE EXPANDED DIAGNOSTIC
ROM.FIRST THE REQUIRED CHECK-
WORDS MUST BE INPUT,AND THE
STARTING LOCATION IN MEMORY.
CHECKSUMS AND CHECKWORD
VERIFICATION CONFIRMS GOOD
DATA IN ROMS.
TO TEST THE EPROM IN THE
SOCKETS. TEST PROCEDURE IS AS
IN SUBTEST 2.
TO TEST SYSTEM ROM. SAME
TEST PROCEDURE AS IN SUBTEST 2.
TO TEST SYSTEM EPROM. SAME
TEST PROCEDURE AS IN SUBTEST 2.

358	002000		SVC
359		000000	SVCINS=0
360		000000	SVCGBL=0
361		000000	SVCTAG=0
362			.TITLE PROGRAM HEADER AND TABLES
363			.SBTTL IDENTIFICATION
364			
365			
366			.SBTTL PROGRAM HEADER
367			
368	002000		BGNMOD MDHEDR
369	002000		MDHEDR::
370			
371			::++
372			:: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
373			:: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
374			::--
375			
376	002000		POINTER BGNSW,BGNSFT
377			
378			
379	002000		HEADER CVM8AB,B,0,0,0,5
380	002000		L\$NAME::
381	002000	103	.ASCII /C/
382	002001	126	.ASCII /V/
383	002002	115	.ASCII /M/
384	002003	070	.ASCII /8/
385	002004	101	.ASCII /A/
386	002005	102	.ASCII /B/
387	002006	000	.BYTE 0
388	002007	000	.BYTE 0
389	002010		L\$REV::
390	002010	102	.ASCII /B/
391	002011		L\$DEPO::
392	002011	060	.ASCII /0/
393	002012		L\$UNIT::
394	002012	000000	.WORD 0
395	002014		L\$TIML::
396	002014	000005	.WORD 5
397	002016		L\$HPCP::
398	002016	014332	.WORD L\$HARD
399	002020		L\$SPCP::
400	002020	014522	.WORD L\$SOFT
401	002022		L\$HPTP::
402	002022	002132	.WORD L\$HW
403	002024		L\$SPTP::
404	002024	002144	.WORD L\$SW
405	002026		L\$LADP::
406	002026	015134	.WORD L\$LAST
407	002030		L\$STA::
408	002030	000000	.WORD 0
409	002032		L\$CO::
410	002032	000000	.WORD 0
411	002034		L\$EFLG::
412	002034	000000	.WORD 0
413	002036		L\$APT::

414	002036	000000		
415	002040		LSDTP::	.WORD 0
416	002040	002112		
417	002042		L\$EXP1::	.WORD L\$DISPATCH
418	002042	000000		
419	002044		L\$EXP2::	.WORD 0
420	002044	000000		
421	002046		L\$EXP3::	.WORD 0
422	002046	000000		
423	002050		L\$MREV::	.WORD 0
424	002050	002		
425	002051	002		
426	002052		L\$TIM1::	.BYTE C\$REVISION
427	002052	000000		
428	002054		L\$TIMU::	.BYTE C\$EDIT
429	002054	000000		
430	002056		L\$EF::	.WORD 0
431	002056	000000		
432	002060	000000		
433	002062		L\$SPC::	.WORD 0
434	002062	000000		
435	002064		L\$DEVP::	.WORD 0
436	002064	003036		
437	002066		L\$REPP::	.WORD L\$DVTYP
438	002066	000000		
439	002070		L\$DRCT::	.WORD 0
440	002070	002506		
441	002072		L\$DRS::	.WORD L\$DR
442	002072	002512		
443	002074		L\$AUT::	.WORD L\$DRST
444	002074	000000		
445	002076		L\$DUT::	.WORD 0
446	002076	000000		
447	002100		L\$TSTID::	.WORD 0
448	002100	000000		
449	002102		L\$DESC::	.WORD 0
450	002102	000000		
451	002104		L\$ICP::	.WORD 0
452	002104	005034		
453	002106		L\$CCP::	.WORD L\$INIT
454	002106	005200		
455	002110			.WORD L\$CLEAN
456				ENDMOD

457
458
459
460
461
462
463
464 002110
465 002110
466 002110
467 002110 000007
468 002112
469 002112 005260
470 002114 006100
471 002116 006730
472 002120 007462
473 002122 007622
474 002124 010376
475 002126 011526
476 002130
477
478
479
480
481
482
483
484
485
486 002130
487 002130 000004
488 002132
489 002132
490
491
492 002132 000000
493 002134 000100
494 002136 000007
495 002140 007777
496
497 002142
498 002142
499
500
501
502
503
504
505
506
507 002142
508 002142 000010
509 002144
510 002144
511
512

```
.SBTTL DISPATCH TABLE

:++
: THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
: IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
:--

          BGNMOD  DSPCODE
DSPCODE::
          DISPATCH 7
          .WORD    7
L$DISPATCH::
          .WORD    T1
          .WORD    T2
          .WORD    T3
          .WORD    T4
          .WORD    T5
          .WORD    T6
          .WORD    T7
          ENDMOD

.SBTTL DEFAULT HARDWARE P-TABLE

:++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
:--

          BGNHW   DFPTBL
          .WORD   L10000-L$HW/2
L$HW::
DFPTBL::

:DEFAULT VALUES FOR UP TO SIX UNITS
          .WORD   0           :UNIT NUMBER 0
          .WORD   100        :INTERRUPT VECTOR
          .WORD   7           :PRIORITY LEVEL
          .WORD   7777       :ROCKER SWITCH SETTINGS
          ENDPHW
L10000:

.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::
```

513
514
515
516
517
518
519
520
521
522
523
524
525
526
527

002144 017042
002146 020656
002150 065162
002152 161744
002154 124453
002156 113667
002160 056040
002162 044734
002164
002164

L10001:

:THE SOFTWARE P-TABLE IS USED TO STORE THE CHECKWORDS
:FOR THE DIAGNOSTIC ROM WHICH IS TESTED IN TEST 6.
.WORD 17042 :ROMA: PAGE 0,1
.WORD 20656 :ROMB: PAGE 2,3
.WORD 65162 :ROMC: PAGE 4,5
.WORD 161744 :ROMD: PAGE 6,7
.WORD 124453 :ROME: PAGE 10,11
.WORD 113667 :ROMF: PAGE 12,13
.WORD 56040 :ROMG: PAGE 14,15
.WORD 44734 :ROMH: PAGE 16,17
ENDSW

528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545

002164
002164
002164

177520
177524
002164

```
.TITLE GLOBAL AREAS
.SBTTL IDENTIFICATION

.SBTTL GLOBAL EQUATES SECTION

:++
: THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
: ARE USED IN MORE THAN ONE TEST.
:--

      BGNMOD  GLBEQAT
GLBEQAT::
      EQUALS

PCR=177520
LSREG=177524
      ENDMOD
```

546
547
548
549
550
551
552
553 002164
554 002164
555 002164 000000
556 002166 000000
557 002170 000000
558 002172 000000
559 002174 000001
560 002176 000000
561 002200 000000
562 002202 000000
563 002204 000000
564 002206 000000
565 002210 000000
566 002212 000000
567 002214 000100
568 002216 000000
569 002220 000000
570 002222 000000
571 002224 000000
572 002226 000000
573 002230 000000
574 002232 000001
575 002234 000000
576 002236 000000
577 002240 000000
578 002242 000000
579 002244 000010
580 002264 000010
581 002304 000100
582 002504
583
584
585
586
587 002504
588 002504 000001
589 002506
590 002506
591 002506 177777
592 002510 000004
593 002512
594 002512
595 002512 000004
596

.SBTTL GLOBAL DATA SECTION

:++
: THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
: IN MORE THAN ONE TEST.
:--

BGNMOD GLBDAT
GLBDAT::
BCF: .WORD 0
REAL: .WORD 0
LOPAG: .WORD 0
COUNTR: .WORD 0
ANSR: .WORD 1
RFLAG: .WORD 0
EXPSUM: .WORD 0
ACTSUM: .WORD 0
PASS: .WORD 0
PASCT: .WORD 0
ULIMIT: .WORD 0
PAGE: .WORD 0
VECT: .WORD 100
SWSET: .WORD 0
STORE: .WORD 0
WORDCT: .WORD 0
PRIOR: .WORD 0
CKWD: .WORD 0
RESPND: .WORD 0
RSET: .WORD 1
LORANG: .WORD 0
HIRANG: .WORD 0
BYTLOC: .WORD 0
ERRFLG: .WORD 0
EXPDIA: .BLKW 10
EPROM: .BLKW 10
SYSROM: .BLKW 100
ENDMOD

: EXPANDED DIAG. ROM CHECKWORDS
: EPROM CHECKWORDS
: SYSTEM ROM/EPROM CHECKWORDS

: STORAGE FOR DEVICE REGISTERS

:
DEVREG 4,177777,DEVDAT,REGMSK
.WORD 1
L\$DR::
REGMSK::
.WORD 177777
.WORD 4
L\$DRST::
DEVDAT::
.BLKW 4

597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652

.SBTTL GLOBAL TEXT SECTION

```

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

:GLOBAL MESSAGES

RWR: .ASCIZ 'READ/WRITE REGISTER ADDRESS: 177522''

PACR: .ASCIZ /PAGE CONTROL REGISTER ADDRESS: 177520/

CKERR: .ASCIZ /CHECKSUM ERROR/

CWDERR: .ASCIZ /INCORRECT CHECKWORD/

LOBYT: .ASCIZ /ERROR OCCURRED IN A LOW BYTE PAGE/

HIBYT: .ASCIZ /ERROR OCCURRED IN A HIGH BYTE PAGE/

LOADR: .ASCIZ /START OF MEMORY RANGE (K)/

.EVEN

```

653
654
655
656      :
657      : NAMES OF DEVICES SUPPORTED BY PROGRAM
658      :
659      :     DEVTYP <BDV11AA>
660      L$DVTYP::
661      :     .ASCIZ /BDV11AA/
662      :     .EVEN
663
664
665      :
666      : FORMAT STATEMENTS USED IN PRINT CALLS
667      :
668
669
670      003046 040445 042522 044507 ZERR: .ASCIZ /%REGISTER CANNOT HOLD ALL ZEROES%/
671      003054 052123 051105 041440
672      003062 047101 047516 020124
673      003070 047510 042114 040440
674      003076 046114 055040 051105
675      003104 042517 022523 000116
676
677      003112 040445 042522 044507 ONERR: .ASCIZ /%REGISTER CANNOT HOLD ALL ONES%/
678      003120 052123 051105 041440
679      003126 047101 047516 020124
680      003134 047510 042114 040440
681      003142 046114 047440 042516
682      003150 022523 000116
683
684      003154 040445 042522 044507 BDDAT: .ASCIZ /%REGISTER CANNOT HOLD GOOD DATA%/
685      003162 052123 051105 041440
686      003170 047101 047516 020124
687      003176 047510 042114 043440
688      003204 047517 020104 040504
689      003212 040524 047045     000
690
691      003217     045 051101 043505 BYTINS: .ASCIZ /%REGISTER IS NOT BYTE ADDRESSABLE%/
692      003224 051511 042524 020122
693      003232 051511 047040 052117
694      003240 041040 052131 020105
695      003246 042101 051104 051505
696      003254 040523 046102 022505
697      003262 000116
698
699      003264 040445 042522 044507 ROT1: .ASCIZ /%REGISTER PICKED UP AN EXTRA SET BIT%/
700      003272 052123 051105 050040
701      003300 041511 042513 020104
702      003306 050125 040440 020116
703      003314 054105 051124 020101
704      003322 042523 020124 044502
705      003330 022524 000116
706
707      003334 040445 042522 044507 RCT0: .ASCIZ /%REGISTER PICKED UP AN EXTRA CLEAR BIT%/
708      003342 052123 051105 050040

```


709 003350 041511 042513 020104
710 003356 050125 040440 020116
711 003364 054105 051124 020101
712 003372 046103 040505 020122
713 003400 044502 022524 000116
714
715 003406 040445 047125 041101
716 003414 042514 052040 020117
717 003422 047514 040503 042524
718 003430 041440 051117 042522
719 003436 052103 046440 046505
720 003444 051117 020131 040520
721 003452 042507 047045 000
722
723 003457 045 046501 046505
724 003464 051117 020131 040522
725 003472 043516 035105 022440
726 003500 031104 040445 026440
727 003506 022440 031104 040445
728 003514 022513 000116
729
730 003520 040445 054105 042520
731 003526 052103 042105 020072
732 003534 047445 022466 032523
733 003542 040445 042522 042503
734 003550 053111 042105 020072
735 003556 047445 022466 000116
736
737

DIAGER: .ASCIZ /%AUNABLE TO LOCATE CORRECT MEMORY PAGE%/

VIRMSG: .ASCIZ /%AMEMORY RANGE: %D2%A - %D2%AK%/

REGDT: .ASCIZ /%AEXPECTED: %06%S5%ARECEIVED: %06%/

.EVEN

```

738
739
740
741
742
743
744
745
746
747
748 003564
749 003564
750 003564
751 003564 012746 003046
752 003570 012746 000001
753 003574 010600
754 003576 104014
755 003600 062706 000004
756 003604
757 003604 010246
758 003606 010146
759 003610 012746 003520
760 003614 012746 000003
761 003620 010600
762 003622 104015
763 003624 062706 000010
764 003630
765 003630
766 003630 104023
767
768 003632
769 003632
770 003632
771 003632 012746 003112
772 003636 012746 000001
773 003642 010600
774 003644 104014
775 003646 062706 000004
776 003652
777 003652 010246
778 003654 010146
779 003656 012746 003520
780 003662 012746 000003
781 003666 010600
782 003670 104015
783 003672 062706 000010
784 003676
785 003676
786 003676 104023
787
788 003700
789 003700
790 003700
791 003700 012746 003154
792 003704 012746 000001
793 003710 010600
    
```

.SBTTL GLOBAL ERROR REPORT SECTION

```

:++
: THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
: THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
: THAT ARE USED BY THE PRINTB AND PRINTX CALLS.
:--
    
```

```

BGNMSG RERR1
RERR1::
PRINTB #ZERR
      MOV #ZERR,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      EMT C$PNTB
      ADD #4,SP
PRINTX #REGDT,R1,R2
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV #REGDT,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      EMT C$PNTX
      ADD #10,SP
ENDMSG
L10002:
      EMT C$MSG

BGNMSG RERR2
RERR2::
PRINTB #ONERR
      MOV #ONERR,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      EMT C$PNTB
      ADD #4,SP
PRINTX #REGDT,R1,R2
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV #REGDT,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      EMT C$PNTX
      ADD #10,SP
ENDMSG
L10003:
      EMT C$MSG

BGNMSG RERR3
RERR3::
PRINTB #RDDAT
      MOV #RDDAT,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
    
```


794	003712	104014		EMT	C\$PNTB
795	003714	062706	000004	ADD	#4,SP
796	003720			PRINTX	#REGDT,R1,R2
797	003720	010246		MOV	R2,-(SP)
798	003722	010146		MOV	R1,-(SP)
799	003724	012746	003520	MOV	#REGDT,-(SP)
800	003730	012746	000003	MOV	#3,-(SP)
801	003734	010600		MOV	SP,R0
802	003736	104015		EMT	C\$PNTX
803	003740	062706	000010	ADD	#10,SP
804	003744			ENDMSG	
805	003744			L10004:	
806	003744	104023		EMT	C\$MSG
807					
808	003746			BGNMSG	RERR4
809	003746			RERR4::	
810	003746			PRINTB	#BYTINS
811	003746	012746	003217	MOV	#BYTINS,-(SP)
812	003752	012746	000001	MOV	#1,-(SP)
813	003756	010600		MOV	SP,R0
814	003760	104014		EMT	C\$PNTB
815	003762	062706	000004	ADD	#4,SP
816	003766			PRINTX	#REGDT,R1,R2
817	003766	010246		MOV	R2,-(SP)
818	003770	010146		MOV	R1,-(SP)
819	003772	012746	003520	MOV	#REGDT,-(SP)
820	003776	012746	000003	MOV	#3,-(SP)
821	004002	010600		MOV	SP,R0
822	004004	104015		EMT	C\$PNTX
823	004006	062706	000010	ADD	#10,SP
824	004012			ENDMSG	
825	004012			L10005:	
826	004012	104023		EMT	C\$MSG
827					
828	004014			BGNMSG	RERR5
829	004014			RERR5::	
830	004014			PRINTB	#ROT1
831	004014	012746	003264	MOV	#ROT1,-(SP)
832	004020	012746	000001	MOV	#1,-(SP)
833	004024	010600		MOV	SP,R0
834	004026	104014		EMT	C\$PNTB
835	004030	062706	000004	ADD	#4,SP
836	004034			ENDMSG	
837	004034			L10006:	
838	004034	104023		EMT	C\$MSG
839					
840	004036			BGNMSG	RERR6
841	004036			RERR6::	
842	004036			PRINTB	#ROTO
843	004036	012746	003334	MOV	#ROTO,-(SP)
844	004042	012746	000001	MOV	#1,-(SP)
845	004046	010600		MOV	SP,R0
846	004050	104014		EMT	C\$PNTB
847	004052	062706	000004	ADD	#4,SP
848	004056			ENDMSG	
849	004056			L10007:	

850	004056	104023		EMT	C\$MSG
851					
852	004060			BGNMSG	PAGERR
853	004060			PAGERR:	
854	004060			PRINTB	#DIAGER
855	004060	012746	003406	MOV	#DIAGER, -(SP)
856	004064	012746	000001	MOV	#1, -(SP)
857	004070	010600		MOV	SP, R0
858	004072	104014		EMT	C\$PNTB
859	004074	062706	000004	ADD	#4, SP
860	004100			ENDMSG	
861	004100			L10010:	
862	004100	104023		EMT	C\$MSG
863					
864					
865					
866	004102			VIPRI:	PRINTF #VIRMSG, LORANG, HIRANG
867	004102	013746	002236	MOV	HIRANG, -(SP)
868	004106	013746	002234	MOV	LORANG, -(SP)
869	004112	012746	003457	MOV	#VIRMSG, -(SP)
870	004116	012746	000003	MOV	#3, -(SP)
871	004122	010600		MOV	SP, R0
872	004124	104017		EMT	C\$PNTF
873	004126	062706	000010	ADD	#10, SP
874					
875				.EVEN	
876					
877					
878					
879					
880					

881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908

.SBTTL GLOBAL SUBROUTINES SECTION

;++
: THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
: THAT ARE USED IN MORE THAN ONE TEST.
:--

;++
: FUNCTIONAL DESCRIPTION:
: SUBROUTINE TO COMPUTE A CHECKSUM IN A ROM/EPROM
: INPUT: CONTENTS OF BCF
: IMPLICIT INPUTS: CONTENTS OF PCR
: OUTPUT: A CHECKSUM VALUE STORED IN LOCATION ACTSUM
: CALLING SEQUENCE: JSR PC,CHKSUM
:--

004132 012701 173776
004136 063701 002164
004142 005037 002202
004146 012702 173000
004152 063702 002164
004156 111204
004160 060437 002202
004164 062702 000002
004170 020201
004172 002771
004174 000207

CHKSUM: MOV #173776,R1 ;STORE THE HIGHEST ADDRESS IN THE ROM
ADD BCF,R1 ;FOR EITHER LOW OR HIGH BYTES
CLR ACTSUM ;CLEAR LOCATION WHICH WILL HOLD THE CHECKSUM
MOV #173000,R2 ;COMPUTE THE LOWEST ADDRESS IN THE ROM
ADD BCF,R2 ;WHERE THE DATA WILL START
1\$: MOVB (R2),R4 ;GET DATA IN BYTES
ADD R4,ACTSUM ;ADD CONTENTS OF EACH LOCATION TO THE CHECKSUM
ADD #2,R2 ;ADJUST ADDRESS
CMP R2,R1 ;COMPARE CURRENT ADDRESS WITH HIGHEST ADDRESS
BLT 1\$;BR IF LESS THAN
RTS PC ;RETURN

909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953

004176
004176 012746 004254
004202 012746 000001
004206 010600
004210 104017
004212 062706 000004
004216
004216 104043
004220 000406
004222 002220
004224 000022
004226 004342
004230 177777
004232 000000
004234 177777
004236
004236 013722 002220
004242 005337 002222
004246 001401
004250 000762
004252 000207
004254 040445 054524 042520
004262 044440 020116 044124
004270 020105 044103 041505
004276 053513 051117 051504
004304 040440 020123 044514
004312 052123 042105 044440
004320 020116 044124 020105
004326 051120 047111 020124
004334 042523 022524 000116
004342 044103 041505 053513
004350 051117 035104 000040

```

:++
:SUBROUTINE TO INPUT CHECKWORDS FROM THE OPERATOR
:INPUTS: NUMBER OF CHECKWORDS TO INPUT
:        POINTER TO STORAGE AREA
:OUTPUTS: CHECKWORDS STORED IN PROPER TABLE
:CALLING SEQUENCE: JSR PC,INPUT
:--

INPUT: PRINTF #INSTR ;PRINT INSTRUCTIONS
      MOV #INSTR,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      EMT C$PNTF
      ADD #4,SP
INLP: GMANID INWORD,STORE,0,-1,0,177777,NO
      EMT C$GMAN
      BR 10000$
      .WORD STORE
      .WORD T$CODE
      .WORD INWORD
      .WORD -1
      .WORD T$LOLIM
      .WORD T$HILIM
10000$:
      MOV STORE,(R2)+ ;PUT CHECKWORD IN TABLE
      DEC WORDCT ;DECREMENT WORD COUNT
      BEQ 1$ ;BR IF FINISHED
      BR INLP ;LOOP UNTIL TABLE IS COMPLETE
      RTS PC ;RETURN
1$:

INSTR: .ASCIZ /%ATYPE IN THE CHECKWORDS AS LISTED IN THE PRINT SET%/

INWORD: .ASCIZ /CHECKWORD: /

      .EVEN
```



```

954      :++
955      :SUBROUTINE TO COMPUTE THE VIRTUAL ADDRESS OF A BAD
956      :PAGE IN MEMORY
957      :INPUTS: PAGE IN PAGE CONTROL REGISTER
958      :        BYTE CONTROL FLAG (BCF)
959      :OUTPUTS: MEMORY RANGE IN WHICH ERROR OCCURRED
960      :CALLING SEQUENCE: JSR PC,VIRTAD
961      :--
962
963 004356 005001      VIRTAD: CLR      R1          :START AT BOTTOM OF RANGE
964 004360 012737 000007 002210      MOV      #7,ULIMIT      :SET UPPER LIMIT OF PAGE
965 004366 113737 177520 002212      MOVB    PCR,PAGE       :LOW PAGE ERROR
966 004374 023737 002212 002210      LPADD:  CMP      PAGE,ULIMIT  :IS PAGE <=ULIMIT
967 004402 003427      BLE      OUTPUT        :BR IF YES
968 004404 022737 000057 002210      CMP      #57,ULIMIT    :IS ULIMIT = 57
969 004412 001006      BNE      1$           :BR IF NO
970 004414 012737 000207 002210      MOV      #207,ULIMIT   :CHANGE UPPER LIMIT
971 004422 012701 000020      MOV      #20,R1        :ADJUST MEMORY POINTER
972 004426 000762      BR      LPADD         :CHECK PAGE AGAIN
973 004430 062737 000010 002210      1$:    ADD      #10,ULIMIT  :INCREASE UPPER LIMIT
974 004436 022737 000377 002210      CMP      #377,ULIMIT  :HAS THE UPPER LIMIT EXCEEDED THE MAX. PAGE
975 004444 002003      BGE      2$           :BR IF NO
976 004446      ERRDF    40,,PAGERR  :COULD NOT FIND THE PAGE OF MEMORY
977 004446 104442      TRAP    T$ERCODE
978 004450 000050      .WORD   40
979 004452 004060      .WORD   PAGERR
980 004454      2$:    CKLOOP
981 004454 104006      EMT     C$CLP1
982 004456 005201      INC     R1            :ADJUST POINTER
983 004460 000745      BR     LPADD         :LOOP UNTIL UPPER LIMIT IS FOUND
984 004462 010137 002234      OUTPUT: MOV     R1,LORANG  :PULL THE LOW RANGE OUT OF THE TABLE
985 004466 013737 002234 002236      MOV     LORANG,HIRANG :COPY THE DATA
986 004474 005237 002236      INC     HIRANG       :INCREMENT TO OBTAIN 1K RANGE
987 004500 015737 002176      TST    RFLAG        :IS IT ROM (2K SEGMENTS)
988 004504 0C1402      BEQ    3$           :BR IF NO
989 004506 0C5237 002236      INC     HIRANG       :OBTAIN 2K RANGE
990 004512 0C0207      3$:    RTS     PC      :RETURN
991

```

```

992      :++
993      :SUBROUTINE TO VERIFY THE CHECKSUM VALUE OF A PAGE
994      :OF EXISTENT MEMORY AND ALSO TEST FOR THE PROPER CHECKWORD.
995      :INPUTS: PAGE CONTROL REGISTER, PAGE CHECKWORD.
996      :OUTPUTS: ERROR FLAGS WHICH POINT TO THE PROPER ERROR MESSAGE
997      :SUBORDINATE ROUTINES USED: CHKSUM
998      :CALLING SEQUENCE: JSR PC, MEMTST
999      :--
1000
1001 004514 005037 002166      MEMTST: CLR      REAL      :CLEAR MEMORY INDICATOR
1002 004520 005037 002164      LOBYTE: CLR      BCF      :SIGNAL LOW BYTES ARE BEING CHECKED
1003 004524 122737 177777 173774      CMPB     #-1,@#173774 :DOES THE ROM EXIST
1004 004532 001421      BEQ      HIBYTE    :BR IF NO
1005 004534 005237 002166      INC      REAL      :INDICATE THAT MEMORY EXISTS
1006 004540 004737 004132      JSR      PC,CHKSUM :COMPUTE THE ACTUAL CHECKSUM
1007 004544 113737 173776 002200      MOVB     @#173776,EXPSUM :GET THE STORED CHECKSUM
1008 004552 063737 002202 002200      ADD      ACTSUM,EXPSUM :ADD THE EXPECTED AND ACTUAL CHECKSUMS
1009 004560 105737 002200      TSTB     EXPSUM     :TEST RESULTING CHECKBYTE
1010 004564 001404      BEQ      1$       :BR IF NO ERROR
1011 004566 012737 000001 002242      MOV      #1,ERRFLG :SET CHECKSUM ERROR FLAG
1012 004574 000207      RTS      PC       :RETURN
1013 004576      1$:
1014
1015 004576 012737 000001 002164      HIBYTE: MOV      #1,BCF      :SET BCF TO DENOTE HIGH BYTES
1016 004604 122737 177777 173775      CMPB     #-1,@#173775 :DOES THE ROM EXIST
1017 004612 001427      BEQ      TSTCKW   :BR IF NO
1018 004614 005737 002166      TST      REAL      :WAS THERE A LOW ROM?
1019 004620 001003      BNE      2$       :BR IF YES
1020 004622 005037 002166      CLR      REAL      :DENOTE NON-EXISTENT LOW ROM
1021 004626 000207      RTS      PC       :RETURN FOR ERROR MESSAGE
1022 004630 005237 002166      2$: INC      REAL      :INDICATE MEMORY EXISTS
1023 004634 004737 004132      JSR      PC,CHKSUM :COMPUTE CHECKSUM
1024 004640 113737 173777 002200      MOVB     @#173777,EXPSUM :GET EXPECTED CHECKSUM
1025 004646 063737 002202 002200      ADD      ACTSUM,EXPSUM :ADD THE EXPECTED AND ACTUAL CHECKSUMS
1026 004654 105737 002200      TSTB     EXPSUM     :TEST RESULTING CHECKBYTE
1027 004660 001404      BEQ      TSTCKW   :BR IF EQUAL
1028 004662 012737 000001 002242      MOV      #1,ERRFLG :SET CHECKSUM ERROR FLAG
1029 004670 000207      RTS      PC       :RETURN
1030
1031 004672 005737 002166      TSTCKW: TST      REAL      :ANY MEMORY?
1032 004676 001420      BEQ      5$       :BR IF NO
1033 004700 022737 000001 002166      CMP      #1,REAL    :SINGLE ROM?
1034 004706 001005      BNE      3$       :BR IF NO
1035 004710 123737 002226 173776      CMPB     CKWD,@#173776 :COMPARE CHECKBYTE ONLY
1036 004716 001005      BNE      4$       :BR IF ERROR
1037 004720 000207      RTS      PC       :RETURN -- NO ERROR
1038 004722 023737 002226 173776      3$: CMP      CKWD,@#173776 :COMPARE CHECKWORD
1039 004730 001403      BEQ      5$       :BR IF NO ERROR
1040 004732 012737 000002 002242      4$: MOV      #2,ERRFLG :DENOTE CHECKSUM ERROR
1041 004740 000207      5$: RTS      PC       :RETURN
1042

```


1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092

004742	013701	002220
004746	020127	000005
004752	003006	
004754	000241	
004756	006101	
004760	006101	
004762	006101	
004764	110104	
004766	000413	
004770	012704	000020
004774	012705	000200
005000	020104	
005002	001404	
005004	005204	
005006	062705	000010
005012	000772	
005014	010504	
005016	110437	002170
005022	005204	
005024	110437	002171
005030	000207	

```

:++
:SUBROUTINE TO COMPUTE THE ACTUAL STARTING PAGE
:OF MEMORY IN WHICH THE MEMORY CHIP IS TO BE
:ADDRESSED.
:INPUTS: THE LOW NUMBER IN THE MEMORY RANGE
:        (I.E. X IN X-Y K)
:OUTPUT: PAGE NUMBER IN PCR WHICH DENOTES WHERE TESTING
:        SHOULD BEGIN.
:CALLING SEQUENCE: JSR PC,SETADR
:--

```

```

SETADR: MOV     STORE,R1      ;COPY DATA
        CMP     R1,#5       ;IS THE NUMBER <=5?
        BGT    1$          ;BR IF NO
        CLC                    ;CLEAR C-BIT FOR ROTATE
        ROL    R1           ;ROTATE TO MULTIPLY
        ROL    R1           ;   BY 10 (8)
        ROL    R1           ;
        MOVB   R1,R4        ;COPY DATA
        BR     LOAD        ;LOAD THE PCR
1$:     MOV     #20,R4      ;START WITH 16 (10)
        MOV     #200,R5    ;CORRESPONDIGE PAGE IS 200
LOOP:  CMP     R1,R4       ;PAGE FOUND?
        BEQ    2$          ;BR IF YES
        INC    R4          ;NEXT PAGE
        ADD    #10,R5      ;NEXT PAGE
        BR     LOOP        ;LOOP UNTIL PAGE IS FOUND
2$:     MOV     R5,R4      ;GET PAGE FOR PCR
LOAD:  MOVB   R4,LOPAG    ;LOW STARTING PAGE
        INC    R4          ;INCREMENT
        MOVB   R4,LOPAG+1 ;HIGH STARTING PAGE
        RTS    PC

```

```

1093 .TITLE MISCELLANEOUS SECTIONS
1094 .SBTTL IDENTIFICATION
1095
1096
1097 .SBTTL REPORT CODING SECTION
1098
1099         BGNRPT
1100 L$RPT::
1101         ENDRPT
1102
1103 L10011:      EMT      C$RPT
1104
1105
1106 .SBTTL INITIALIZE SECTION
1107
1108 :++
1109 : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
1110 : AT THE BEGINNING OF EACH PASS.
1111 :--
1112
1113         BGNINIT
1114 L$INIT::
1115         GPWARD #0,R1          ;GET POINTER TO BASE ADDRESS OF P-TABLE
1116         MOV    #0,R0
1117         EMT    C$GPHRD
1118         MOV    R0,R1
1119         MOV    2(R1),VECT      ;GET INTERRUPT VECTOR
1120         MOV    4(R1),PRIOR    ;GET PRIORITY LEVEL
1121         MOV    6(R1),SWSET    ;GET ROCKER SWITCH SETTINGS
1122         SETPRI #PRI07        ;INHIBIT INTERRUPTS
1123         MOV    #PRI07,R0
1124         EMT    C$SPRI
1125
1126         MANUAL          ;MANUAL INTERVENTION OK?
1127         EMT    C$MANI
1128         BNCOMPLETE     OUT    ;BR IF NO
1129         BCC     OUT
1130         PRINTF #IDENT      ;PRINT PROGRAM I.D.
1131         MOV    #IDENT,-(SP)
1132         MOV    #1,-(SP)
1133         MOV    SP,R0
1134         EMT    C$PNTF
1135         ADD    #4,SP
1136
1137         OUT:
1138         EXIT    INIT
1139         EMT    C$EXIT
1140         .WORD  L10012-.
1141
1142 IDENT:  .ASCIZ  '%ABDV11-AA BOOTSTRAP DIAGNOSTIC PROGRAM%N'
1143
1144
1145
1146
1147
1148 .EVEN
    
```


1149
1150 005176
1151 005176
1152 005176 104011

L10012: ENDINIT
EMT C\$INIT

```

1153      .SBTTL  CLEANUP CODING SECTION
1154
1155      :++
1156      : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
1157      : AT THE END OF EACH PASS.
1158      :--
1159
1160      005200      BGNCLN
1161      005200      L$CLEAN::
1162
1163      005200      005037      177520      CLR      PCR      ;CLEAR PAGE CONTROL REGISTER
1164      005204      005037      177522      CLR      RWREG    ;CLEAR READ/WRITE REGISTER
1165      005210      012737      000001      002232      MOV      #1,RSET  ;RESTORE DEFAULT VALUE
1166      005216      005037      002230      CLR      RESPND   ;RESTORE DEFAULT
1167      005222      005037      013540      CLR      ADDON    ;RESTORE DEFAULT
1168      005226      012737      000001      002174      MOV      #1,ANSR  ;RESTORE DEFAULT
1169      005234      005237      002204      INC      PASS     ;INCREMENT PASS COUNT
1170      005240      005237      002206      INC      PASCT    ;INCREMENT TEST 4 PASS COUNT
1171      005244      CLRVEC   VECT      ;CLEAR INTERRUPT VECTOR
1172      005244      013700      002214      MOV      VECT,R0
1173      005250      104036      EMT      C$CVEC
1174
1175      005252      EXIT      CLN
1176      005252      104032      EMT      C$EXIT
1177      005254      000002      .WORD   L10013-.
1178
1179
1180
1181      005256      ENDCLN
1182      005256      L10013:
1183      005256      104012      EMT      C$CLEAN
1184

```


1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197 005260
 1198
 1199 005260
 1200 005260 104002
 1201 005262 005037 177522
 1202 005266 001411
 1203 005270 005001
 1204 005272 013702 177522
 1205 005276
 1206 005276 104562
 1207 005300 000001
 1208 005302 002522
 1209 005304 003564
 1210 005306
 1211 005306 104032
 1212 005310 000566
 1213 005312
 1214 005312 104006
 1215 005314
 1216 005314
 1217 005314 104003
 1218
 1219 005316
 1220 005316 104002
 1221 005320 012737 177777 177522
 1222 005326 022737 177777 177522
 1223 005334 001412
 1224 005336 012701 177777
 1225 005342 013702 177522
 1226 005346
 1227 005346 104562
 1228 005350 000002
 1229 005352 002522
 1230 005354 003632
 1231 005356
 1232 005356 104032
 1233 005360 000516
 1234 005362
 1235 005362 104006
 1236 005364
 1237 005364
 1238 005364 104003
 1239
 1240 005366

.TITLE HARDWARE TESTS
 .SBTTL IDENTIFICATION

.SBTTL TEST 1: READ/WRITE REGISTER TEST

:++
 :TEST TO VERIFY THAT THE READ/WRITE REGISTER AT ADDRESS 177522
 :IS WORD AND BYTE ADDRESSABLE.
 :--

RWREG=177522

BGNTST

BGNSUB

EMT CSBSUB

CLR RWREG

BEQ 1\$

CLR R1

MOV RWREG,R2

ERRDF 1,RWR,RERR1,CKLOOP

TRAP T\$ERCODE

.WORD 1

.WORD RWR

.WORD RERR1

EXIT TST

EMT C\$EXIT

.WORD L10014-

1\$: CKLOOP

EMT C\$CLP1

ENDSUB

L10015:

EMT C\$ESUB

BGNSUB

EMT CSBSUB

MOV #-1,RWREG

CMP #177777,RWREG

BEQ 2\$

MOV #-1,R1

MOV RWREG,R2

ERRDF 2,RWR,RERR2,CKLOOP

TRAP T\$ERCODE

.WORD 2

.WORD RWR

.WORD RERR2

EXIT TST

EMT C\$EXIT

.WORD L10014-

2\$: CKLOOP

EMT C\$CLP1

ENDSUB

L10016:

EMT C\$ESUB

BGNSUB

:LOAD ALL ZEROS

:BR IF CLEAR

:EXPECTED DATA

:COPY CONTENTS

:REGISTER CANNOT HOLD ALL ZEROS

:ABORT TEST IF LOOP ON ERROR NOT SELECTED

:LOOP ON ERROR IF SELECTED

:LOAD ALL ONES

:CHECK THE REGISTER

:BR IF HOLDING GOOD DATA

:EXPECTED DATA

:COPY CONTENTS

:REGISTER CANNOT HOLD ALL ONES

:ABORT TEST IF ERROR AND NO LOOPING

:LOOP ON ERROR IF SELECTED

1241	005366	104002			EMT	C\$BSUB	
1242	005370	012737	125252	177522	MOV	#125252,RWREG	;LOAD ALTERNATING 1'S AND 0'S BIT PATTERN
1243	005376	022737	125252	177522	CMP	#125252,RWREG	;CHECK DATA
1244	005404	001412			BEQ	3\$;BR IF GOOD
1245	005406	012701	125252		MOV	#125252,R1	;EXPECTED DATA
1246	005412	013702	177522		MOV	RWREG,R2	;COPY CONTENTS
1247	005416				ERRDF	3,RWR,RERR3,CKLOOP	;CANNOT HOLD GOOD DATA
1248	005416	104562			TRAP	T\$ERCODE	
1249	005420	000003			.WORD	3	
1250	005422	002522			.WORD	RWR	
1251	005424	003700			.WORD	RERR3	
1252	005426				EXIT	TST	;ABORT TEST IF ERROR DETECTED
1253	005426	104032			EMT	C\$EXIT	
1254	005430	000446			.WORD	L10014-	
1255	005432		3\$:		CKLOOP		;CHECK FOR LOOP ON ERROR AGAIN
1256	005432	104006			EMT	C\$CLP1	
1257	005434				ENDSUB		
1258	005434			L10017:			
1259	005434	104003			EMT	C\$ESUB	
1260							
1261	005436				BGNSUB		
1262	005436	104002			EMT	C\$BSUB	
1263	005440	105037	177522		CLRB	RWREG	;CLEAR THE REGISTER'S LOW BYTE
1264	005444	022737	125000	177522	CMP	#125000,RWREG	;DID IT CLEAR PROPERLY?
1265	005452	001412			BEQ	4\$;BR IF YES
1266	005454	012701	125000		MOV	#125000,R1	;EXPECTED DATA
1267	005460	013702	177522		MOV	RWREG,R2	;COPY CONTENTS
1268	005464				ERRDF	4,RWR,RERR4,CKLOOP	;DID NOT RESPOND PROPERLY TO BYTE INSTRUCTION
1269	005464	104562			TRAP	T\$ERCODE	
1270	005466	000004			.WORD	4	
1271	005470	002522			.WORD	RWR	
1272	005472	003746			.WORD	RERR4	
1273	005474				EXIT	TST	;ABORT TEST IF ERROR DETECTED
1274	005474	104032			EMT	C\$EXIT	
1275	005476	000400			.WORD	L10014-	
1276	005500		4\$:		CKLOOP		;CHECK FOR LOOP ON ERROR AGAIN
1277	005500	104006			EMT	C\$CLP1	
1278	005502				ENDSUB		
1279	005502			L10020:			
1280	005502	104003			EMT	C\$ESUB	
1281							
1282	005504				BGNSUB		
1283	005504	104002			EMT	C\$BSUB	
1284	005506	000337	177522		SWAB	RWREG	;SWAP BYTES IN THE REGISTER
1285	005512	022737	000252	177522	CMP	#252,RWREG	;GOOD DATA?
1286	005520	001406			BEQ	5\$;BR IF YES
1287	005522				ERRDF	5,RWR,RERR4,CKLOOP	;BYTE INSTRUCTION ERROR
1288	005522	104562			TRAP	T\$ERCODE	
1289	005524	000005			.WORD	5	
1290	005526	002522			.WORD	RWR	
1291	005530	003746			.WORD	RERR4	
1292	005532				EXIT	TST	;ABORT TEST IF ERROR DETECTED
1293	005532	104032			EMT	C\$EXIT	
1294	005534	000342			.WORD	L10014-	
1295	005536		5\$:		CKLOOP		;CHECK FOR LOOP ON ERROR AGAIN
1296	005536	104006			EMT	C\$CLP1	


```

1297 005540          ENDSUB
1298 005540          L10021: EMT      C$ESUB
1299 005540 104003   EMT      C$ESUB
1300
1301 005542          BGNSUB
1302 005542 104002   EMT      C$BSUB
1303 005544 012737 052525 177522  MOV     #052525,RWREG      ;LOAD AN ALTERNATING 0'S AND 1'S BIT PATTERN
1304 005552 022737 052525 177522  CMP     #052525,RWREG      ;CHECK IT
1305 005560 001412   BEQ     6$                ;BR IF GOOD DATA
1306 005562 012701 052525   MOV     #052525,R1        ;EXPECTED DATA
1307 005566 013702 177522   MOV     RWREG,R2         ;COPY CONTENTS
1308 005572          ERRDF   6,RWR,RERR3,CKLOOP    ;CANNOT HOLD GOOD DATA
1309 005572 104562   TRAP   T$ERCODE
1310 005574 000006   .WORD  6
1311 005576 002522   .WORD  RWR
1312 005600 003700   .WORD  RERR3
1313 005602          EXIT     TST                ;ABORT TEST IF ERROR DETECTED
1314 005602 104032   EMT     C$EXIT
1315 005604 000272   .WORD  L10014-.
1316 005606          6$:   CKLOOP
1317 005606 104006   EMT     C$CLP1            ;CHECK FOR LOOP ON ERROR AGAIN
1318 005610          ENDSUB
1319 005610          L10022:
1320 005610 104003   EMT     C$ESUB
1321
1322 005612          BGNSUB
1323 005612 104002   EMT     C$BSUB
1324 005614 105037 177523   CLRB   RWREG+1           ;CLEAR HIGH BYTE OF REGISTER
1325 005620 022737 000125 177522  CMP     #125,RWREG        ;CHECK THE RESULTING CONTENTS OF THE REGISTER
1326 005626 001412   BEQ     7$                ;BR IF GOOD DATA
1327 005630 012701 000125   MOV     #125,R1          ;EXPECTED DATA
1328 005634 013702 177522   MOV     RWREG,R2         ;COPY CONTENTS
1329 005640          ERRDF   7,RWR,RERR4,CKLOOP    ;BYTE INSTRUCTION ERROR
1330 005640 104562   TRAP   T$ERCODE
1331 005642 000007   .WORD  7
1332 005644 002522   .WORD  RWR
1333 005646 003746   .WORD  RERR4
1334 005650          EXIT     TST                ;ABORT TEST IF ERROR DETECTED
1335 005650 104032   EMT     C$EXIT
1336 005652 000224   .WORD  L10014-.
1337 005654          7$:   CKLOOP
1338 005654 104006   EMT     C$CLP1            ;CHECK FOR LOOP ON ERROR AGAIN
1339 005656          ENDSUB
1340 005656          L10023:
1341 005656 104003   EMT     C$ESUB
1342
1343 005660          BGNSUB
1344 005660 104002   EMT     C$BSUB
1345 005662 000337 177522   SWAB   RWREG              ;SWAP BYTES
1346 005666 022737 052400 177522  CMP     #052400,RWREG      ;DATA GOOD?
1347 005674 001412   BEQ     10$               ;BR IF YES
1348 005676 012701 052400   MOV     #52400,R1         ;EXPECTED DATA
1349 005702 013702 177522   MOV     RWREG,R2         ;COPY CONTENTS
1350 005706          ERRDF   10,RWR,RERR4,CKLOOP   ;BYTE INSTRUCTION ERROR
1351 005706 104562   TRAP   T$ERCODE
1352 005710 000012   .WORD  10
    
```

```

1353 005712 002522          .WORD RWR
1354 005714 003746          .WORD RERR4
1355 005716                EXIT TST          ;ABORT TEST IF ERROR DETECTED
1356 005716 104032          EMT C$EXIT
1357 005720 000156          .WORD L10014-.
1358 005722                10$: CKLOOP          ;CHECK FOR LOOP ON ERROR AGAIN
1359 005722 104006          EMT C$CLP1
1360 005724                ENDSUB
1361 005724                L10024:
1362 005724 104003          EMT C$ESUB
1363
1364 005726                BGNSUB
1365 005726 104002          EMT C$BSUB
1366 005730 005037 177522    CLR RWREG          ;MAKE SURE THE C-BIT IS CLEAR
1367 005734 052737 100000 177522  BIS #BIT15,RWREG  ;SET MSB
1368 005742 013703 177522    MOV RWREG,R3      ;COPY DATA IN RWREG
1369 005746 023703 177522    ROTLP1: CMP RWREG,R3 ;ARE THEY THE SAME?
1370 005752 001005          BNE 11$           ;BR IF NO
1371 005754 006003          ROR R3            ;ROTATE THE SET BIT
1372 005756 001411          BEQ 12$           ;BR WHEN FINISHED
1373 005760 006037 177522    ROR RWREG         ;REPEAT ROTATE
1374 005764 000770          BR ROTLP1        ;LCOP UNTIL ROTATE IS COMPLETE
1375 005766                11$: ERRDF 11,RWR,RERR5,CKLOOP
1376 005766 104562          TRAP T$ERCODE
1377 005770 000013          .WORD 11
1378 005772 002522          .WORD RWR
1379 005774 004014          .WORD RERR5
1380 005776                EXIT TST          ;SKIP REST OF TEST
1381 005776 104032          EMT C$EXIT
1382 006000 000076          .WORD L10014-.
1383 006002                12$: CKLOOP
1384 006002 104006          EMT C$CLP1        ;CHECK FOR LOOP ON ERROR
1385 006004                ENDSUB
1386 006004                L10025:
1387 006004 104003          EMT C$ESUB
1388
1389 006006                BGNSUB
1390 006006 104002          EMT C$BSUB
1391 006010 012737 177777 177522  MOV #-1,RWREG     ;SET ALL ONES
1392 006016 042737 100000 177522  BIC #BIT15,RWREG  ;CLEAR MSB
1393 006024 013703 177522    MOV RWREG,R3      ;COPY DATA
1394 006030 023703 177522    ROTLP2: CMP RWREG,R3 ;ARE THEY THE SAME?
1395 006034 001010          BNE 13$           ;BR IF NO
1396 006036 000261          SEC              ;SET C-BIT FOR ROTATE
1397 006040 006037 177522    ROR RWREG         ;ROTATE CLEAR BIT
1398 006044 006003          ROR R3            ;REPEAT
1399 006046 022703 077777    CMP #077777,R3    ;FINISHED?
1400 006052 001366          BNE ROTLP2        ;BR IF NOT YET
1401 006054 000406          BR 14$           ;SUBTEST FINISHED
1402 006056                13$: ERRDF 12,RWR,RERR6,CKLOOP
1403 006056 104562          TRAP T$ERCODE
1404 006060 000014          .WORD 12
1405 006062 002522          .WORD RWR
1406 006064 004036          .WORD RERR6
1407 006066                EXIT TST
1408 006066 104032          EMT C$EXIT
    
```


1409	006070	000006		
1410	006072		14\$:	.WORD L10014-
1411	006072	104006		CKLOOP
1412	006074			EMT C\$CLP1
1413	006074		L10026:	ENDSUB
1414	006074	104003		EMT C\$ESUB
1415				
1416	006076			ENDTST
1417	006076		L10014:	
1418	006076	104001		EMT C\$ETST

```

1419
1420
1421
1422
1423
1424
1425 006100
1426
1427 006100
1428 006100 104002
1429 006102 005037 177520
1430 006106 001411
1431 006110 005001
1432 006112 013702 177520
1433 006116
1434 006116 104562
1435 006120 000015
1436 006122 002566
1437 006124 003564
1438 006126
1439 006126 104032
1440 006130 000576
1441 006132
1442 006132 104006
1443 006134
1444 006134
1445 006134 104003
1446
1447 006136
1448 006136 104002
1449 006140 012737 177777 177520
1450 006146 022737 177777 177520
1451 006154 001412
1452 006156 012701 177777
1453 006162 013702 177520
1454 006166
1455 006166 104562
1456 006170 000016
1457 006172 002566
1458 006174 003632
1459 006176
1460 006176 104032
1461 006200 000526
1462 006202
1463 006202 104006
1464 006204
1465 006204
1466 006204 104003
1467
1468
1469 006206
1470 006206 104002
1471 006210 012737 125252 177520
1472 006216 022737 125252 177520
1473 006224 001412
1474 006226 012701 125252

```

```

      .SBTTL TEST 2: PAGE CONTROL REGISTER TEST
      :++
      :TEST TO VERIFY THAT THE PAGE CONTROL REGISTER IS WORD
      :AND BYTE ADDRESSABLE.
      :--

      BGNTST

      BGNSUB
      EMT C$BSUB
      CLR PCR ;LOAD ALL ZEROS
      BEQ 1$ ;BR IF CLEARED
      CLR R1 ;EXPECTED DATA
      MOV PCR,R2 ;COPY CONTENTS
      ERRDF 13,PACR,RERR1,CKLOOP ;REGISTER CANNOT HOLD ALL ZEROS
      TRAP T$ERCODE
      .WORD 13
      .WORD PACR
      .WORD RERR1
      EXIT TST ;ABORT TEST IF ERROR DETECTED
      EMT C$EXIT
      .WORD L10027-.
      1$: CKLOOP ;CHECK FOR LOOP ON ERROR AGAIN
      EMT C$CLP1
      ENDSUB
      L10030:
      EMT C$ESUB

      BGNSUB
      EMT C$BSUB
      MOV #-1,PCR ;LOAD ALL ONES
      CMP #177777,PCR ;CHECK FOR GOOD DATA
      BEQ 2$ ;BR IF GOOD
      MOV #-1,R1 ;EXPECTED DATA
      MOV PCR,R2 ;COPY CONTENTS
      ERRDF 14,PACR,RERR2,CKLOOP ;REGISTER CANNOT HOLD ALL ONES
      TRAP T$ERCODE
      .WORD 14
      .WORD PACR
      .WORD RERR2
      EXIT TST ;ABORT TEST IF ERROR DETECTED
      EMT C$EXIT
      .WORD L10027-.
      2$: CKLOOP ;CHECK FOR LOOP ON ERROR AGAIN
      EMT C$CLP1
      ENDSUB
      L10031:
      EMT C$ESUB

      BGNSUB
      EMT C$BSUB
      MOV #125252,PCR ;LOAD AN ALTERNATING 1'S AND 0'S BIT PATTERN
      CMP #125252,PCR ;CHECK THE RESULTS
      BEQ 3$ ;BR IF GOOD DATA
      MOV #125252,R1 ;EXPECTED DATA

```


1475	006232	013702	177520	MOV	PCR,R2	:COPY CONTENTS
1476	006236			ERRDF	15,PACR,RERR3,CKLOOP	:REGISTER CANNOT HOLD GOOD DATA
1477	006236	104562		TRAP	T\$ERCODE	
1478	006240	000017		.WORD	15	
1479	006242	002566		.WORD	PACR	
1480	006244	003700		.WORD	RERR3	
1481	006246			EXIT	TST	:ABORT TEST IF ERROR DETECTED
1482	006246	104032		EMT	C\$EXIT	
1483	006250	000456		.WORD	L10027-	
1484	006252			3\$: CKLOOP		:CHECK FOR LOOP ON ERROR AGAIN
1485	006252	104006		EMT	C\$CLP1	
1486	006254			ENDSUB		
1487	006254			L10032: EMT	C\$ESUB	
1488	006254	104003				
1489						
1490	006256			BGNSUB		
1491	006256	104002		EMT	C\$BSUB	
1492	006260	105037	177520	CLRB	PCR	:CLEAR THE REGISTER'S LOW BYTE
1493	006264	022737	125000 177520	CMP	#125000,PCR	:COMPARE THE RESULTS
1494	006272	001412		BEQ	4\$:BR IF GOOD DATA
1495	006274	012701	125000	MOV	#125000,R1	:EXPECTED DATA
1496	006300	013702	177520	MOV	PCR,R2	:COPY CONTENTS
1497	006304			ERRDF	16,PACR,RERR4,CKLOOP	:BYTE INSTRUCTION ERROR
1498	006304	104562		TRAP	T\$ERCODE	
1499	006306	000020		.WORD	16	
1500	006310	002566		.WORD	PACR	
1501	006312	003746		.WORD	RERR4	
1502	006314			EXIT	TST	:ABORT TEST IF ERROR DETECTED
1503	006314	104032		EMT	C\$EXIT	
1504	006316	000410		.WORD	L10027-	
1505	006320			4\$: CKLOOP		:CHECK FOR LOOP ON ERROR
1506	006320	104006		EMT	C\$CLP1	
1507	006322			ENDSUB		
1508	006322			L10033: EMT	C\$ESUB	
1509	006322	104003				
1510						
1511	006324			BGNSUB		
1512	006324	104002		EMT	C\$BSUB	
1513	006326	000337	177520	SWAB	PCR	:SWAP BYTES
1514	006332	022737	000252 177520	CMP	#252,PCR	:CHECK THE RESULTS
1515	006340	001412		BEQ	5\$:BR IF GOOD DATA
1516	006342	012701	000252	MOV	#252,R1	:EXPECTED DATA
1517	006346	013702	177520	MOV	PCR,R2	:COPY CONTENTS
1518	006352			ERRDF	17,PACR,RERR4,CKLOOP	:BYTE INSTRUCTION ERROR
1519	006352	104562		TRAP	T\$ERCODE	
1520	006354	000021		.WORD	17	
1521	006356	002566		.WORD	PACR	
1522	006360	003746		.WORD	RERR4	
1523	006362			EXIT	TST	:ABORT TEST IF ERROR DETECTED
1524	006362	104032		EMT	C\$EXIT	
1525	006364	000342		.WORD	L10027-	
1526	006366			5\$: CKLOOP		:CHECK FOR LOOP ON ERROR
1527	006366	104006		EMT	C\$CLP1	
1528	006370			ENDSUB		
1529	006370			L10034: EMT	C\$ESUB	
1530	006370	104003				

1531	006372				BGNSUB		
1532	006372	104002			EMT	C\$BSUB	
1533	006374	012737	052525	177520	MOV	#052525,PCR	:LOAD AN ALTERNATING 0'S AND 1'S BIT PATTERN
1534	006402	022737	052525	177520	CMP	#052525,PCR	:CHECK THE RESULTS
1535	006410	001412			BEQ	6\$:BR IF GOOD DATA
1536	006412	012701	052525		MOV	#052525,R1	:EXPECTED DATA
1537	006416	013702	177520		MOV	PCR,R2	:COPY CONTENTS
1538	006422				ERRDF	20,PACR,RERR3,CKLOOP	:REGISTER CANNOT HOLD GOOD DATA
1539	006422	104562			TRAP	T\$ERCODE	
1540	006424	000024			.WORD	20	
1541	006426	002566			.WORD	PACR	
1542	006430	003700			.WORD	RERR3	
1543	006432				EXIT	TST	:ABORT TEST IF ERROR DETECTED
1544	006432	104032			EMT	C\$EXIT	
1545	006434	000272			.WORD	L10027-	
1546	006436		6\$:		CKLOOP		:CHECK FOR LOOP ON ERROR
1547	006436	104006			EMT	C\$CLP1	
1548	006440				ENDSUB		
1549	006440			L10035:			
1550	006440	104003			EMT	C\$ESUB	
1551	006442				BGNSUB		
1552	006442	104002			EMT	C\$BSUB	
1553	006444	105037	177521		CLRB	PCR+1	:CLEAR THE HIGH BYTE
1554	006450	022737	000125	177520	CMP	#125,PCR	:CHECK THE REGISTER CONTENTS
1555	006456	001412			BEQ	7\$:BR IF GOOD DATA
1556	006460	012701	000125		MOV	#125,R1	:EXPECTED DATA
1557	006464	013702	177520		MOV	PCR,R2	:COPY CONTENTS
1558	006470				ERRDF	21,PACR,RERR4,CKLOOP	:BYTE INSTRUCTION ERROR
1559	006470	104562			TRAP	T\$ERCODE	
1560	006472	000025			.WORD	21	
1561	006474	002566			.WORD	PACR	
1562	006476	003746			.WORD	RERR4	
1563	006500				EXIT	TST	:ABORT TEST IF ERROR DETECTED
1564	006500	104032			EMT	C\$EXIT	
1565	006502	000224			.WORD	L10027-	
1566	006504		7\$:		CKLOOP		:CHECK FOR LOOP ON ERROR
1567	006504	104006			EMT	C\$CLP1	
1568	006506				ENDSUB		
1569	006506			L10036:			
1570	006506	104003			EMT	C\$ESUB	
1571	006510				BGNSUB		
1572	006510	104002			EMT	C\$BSUB	
1573	006510	000337	177520		SWAB	PCR	:SWAP BYTES
1574	006512	000337	052400	177520	CMP	#052400,PCR	:CHECK RESULTING CONTENTS
1575	006516	022737	052400	177520	BEQ	10\$:BR IF GOOD DATA
1576	006524	001412			MOV	#52400,R1	:EXPECTED DATA
1577	006526	012701	052400		MOV	PCR,R2	:COPY CONTENTS
1578	006532	013702	177520		ERRDF	22,PACR,RERR4,CKLOOP	:BYTE INSTRUCTION ERROR
1579	006536				TRAP	T\$ERCODE	
1580	006536	104562			.WORD	22	
1581	006540	000026			.WORD	PACR	
1582	006542	002566			.WORD	RERR4	
1583	006544	003746			EXIT	TST	:ABORT TEST IF ERROR DETECTED
1584	006546				EMT	C\$EXIT	
1585	006546	104032			.WORD	L10027-	
1586	006550	000156					


```

1587 006552      10$:  CKLOOP      ;CHECK FOR LOOP ON ERROR
1588 006552 104006      EMT      C$CLP1
1589 006554      ENDSUB
1590 006554      L10037:
1591 006554 104003      EMT      C$ESUB
1592
1593 006556      BGNSUB
1594 006556 104002      EMT      C$BSUB
1595 006560 005037 177520      CLR      PCR
1596 006564 052737 100000 177520      BIS      #BIT15,PCR      ;MAKE SURE THE C-BIT IS CLEAR
1597 006572 013703 177520      MOV      PCR,R3      ;SET MSB
1598 006576 023703 177520      ROTLP3: CMP      PCR,R3      ;COPY DATA IN PCR
1599 006602 001005      BNE      11$      ;ARE THEY THE SAME?
1600 006604 006003      ROR      R3      ;BR IF NO
1601 006606 001411      BEQ      12$      ;ROTATE THE SET BIT
1602 006610 006037 177520      ROR      PCR      ;BR IF FINISHED
1603 006614 000770      BR       ROTLP3      ;REPEAT ROTATE
1604 006616      11$:  ERRDF      23,PACR,RERR5,CKLOOP ;LOOP UNTIL ROTATE IS COMPLETE
1605 006616 104562      TRAP     T$ERCODE
1606 006620 000027      .WORD   23
1607 006622 002566      .WORD   PACR
1608 006624 004014      .WORD   RERR5
1609 006626      EXIT     TST
1610 006626 104032      EMT      C$EXIT      ;SKIP REST OF TEST
1611 006630 000076      .WORD   L10027-.
1612 006632      12$:  CKLOOP      ;CHECK FOR LOOP ON ERROR
1613 006632 104006      EMT      C$CLP1
1614 006634      ENDSUB
1615 006634      L10040:
1616 006634 104003      EMT      C$ESUB
1617
1618 006636      BGNSUB
1619 006636 104002      EMT      C$BSUB
1620 006640 012737 177777 177520      MOV      #-1,PCR
1621 006646 042737 100000 177520      BIC      #BIT15,PCR      ;SET ALL ONES
1622 006654 013703 177520      MOV      PCR,R3      ;CLEAR MSB
1623 006660 023703 177520      ROTLP4: CMP      PCR,R3      ;COPY DATA
1624 006664 001010      BNE      13$      ;ARE THEY THE SAME?
1625 006666 000261      SEC      ;BR IF NO
1626 006670 006037 177520      ROR      PCR      ;SET C-BIT FOR ROTATE
1627 006674 006003      ROR      R3      ;ROTATE CLEAR BIT
1628 006676 022703 077777      CMP      #077777,R3      ;REPEAT
1629 006702 001366      BNE      ROTLP4      ;ALL ONES?
1630 006704 000406      BR       14$      ;BR IF NOT YET
1631 006706      13$:  ERRDF      24,PACR,RERR6,CKLOOP ;SUBTEST FINISHED
1632 006706 104562      TRAP     T$ERCODE
1633 006710 000030      .WORD   24
1634 006712 002566      .WORD   PACR
1635 006714 004036      .WORD   RERR6
1636 006716      EXIT     TST
1637 006716 104032      EMT      C$EXIT
1638 006720 009006      .WORD   L10027-.
1639 006722      14$:  CKLOOP
1640 006722 104006      EMT      C$CLP1
1641 006724      ENDSUB
1642 006724      L10041:
    
```

1643 006724 104003
1644 006726
1645 006726
1646 006726 104001
1647

EMT C\$ESUB
ENDTST
L10027:
EMT C\$ETST


```
1648 .SBTTL TEST 3: BEVENT CLAMP ENABLE TEST
1649 :++
1650 :TEST TO VERIFY THAT THE BEVENT CLAMP CAN BE ENABLED. THIS TEST
1651 :ASSUMES THAT SWITCH #5 OF E21 IS IN THE ON POSITION, AND THE M8012
1652 :MODULE IS LOCATED IN THE SAME BACKPLANE THAT THE LINE TIME CLOCK
1653 :IS GENERATED FROM.
1654 :--
1655
1656 006730 BGNTST
1657
1658 177546 BEVREG=177546
1659
1660 006730 005737 002204 TST PASS ;IF THIS IS FIRST PASS
1661 006734 001402 BEQ 1$ ;THEN DO THE TEST
1662 006736 EXIT TST ;ELSE DON'T
1663 006736 104032 EMT C$EXIT
1664 006740 000520 .WORD L10042-.
1665 006742 1$: BGNSUB
1666 006742 104002 EMT C$SUB
1667 006744 SETVEC VECT,#INTSR,#PRI07 ;SET INTERRUPT VECTOR,INHIBIT INTERRUPTS
1668 006744 012746 000340 MOV #PRI07,-(SP)
1669 006750 012746 007134 MOV #INTSR,-(SP)
1670 006754 013746 002214 MOV VECT,-(SP)
1671 006760 012746 000003 MOV #3,-(SP)
1672 006764 104037 EMT C$SVEC
1673 006766 062706 000010 ADD #10,SP
1674 006772 052737 000100 177546 BIS #BIT06,BEVREG ;REMOVE BEVENT CLAMP
1675 007000 SETPRI #PRI00 ;ALLOW INTERRUPTS
1676 007000 012700 000000 MOV #PRI00,R0
1677 007004 104041 EMT C$SPRI
1678 007006 WAITUS #400. ;DELAY 40 MSECS.
1679 007006 012700 000620 MOV #400.,R0
1680 007012 104027 EMT C$WTU
1681 007014 SETPRI #PRI07 ;INHIBIT FURTHER INTERRUPTS
1682 007014 012700 000340 MOV #PRI07,R0
1683 007020 104041 EMT C$SPRI
1684 007022 022737 000002 007142 CMP #2,ICOUNT ;DID THE MINIMUM OF TWO INTERRUPTS OCCUR?
1685 007030 003403 BLE 2$ ;BR IF YES
1686 007032 ERRDF 25, BVERR1,CKLOOP ;BEVENT CLAMP ENABLE FAILED
1687 007032 104542 TRAP T$ERCODE
1688 007034 000031 .WORD 25
1689 007036 007144 .WORD BVERR1
1690 007040 2$: CKLOOP ;CHECK FOR LOOP ON ERROR
1691 007040 104006 EMT C$CLP1
1692 007042 005037 007142 CLR ICOUNT ;CLEAR INTERRUPT COUNT
1693 007046 ENDSUB
1694 007046 L10043:
1695 007046 104003 EMT C$ESUB
1696
1697 007050 BGNSUB
1698 007050 104002 EMT C$SUB
1699 007052 042737 000100 177546 BIC #BIT06,BEVREG ;SET BEVENT CLAMP
1700 007060 SETPRI #PRI00 ;ALLOW INTERRUPTS
1701 007060 012700 000000 MOV #PRI00,R0
1702 007064 104041 EMT C$SPRI
1703 007066 WAITUS #400. ;DELAY 40 MSECS.
```

1704	007066	012700	000620	MOV	#400.,R0	
1705	007072	104027		EMT	C\$WTU	
1706	007074			SETPRI	#PRI07	;SET HIGHEST PRIORITY
1707	007074	012700	000340	MOV	#PRI07,R0	
1708	007100	104041		EMT	C\$SPRI	
1709	007102	022737	000001 007142	CMP	#1,ICOUNT	;CHECK INTERRUPT COUNT
1710	007110	002003		BGE	4\$;BR IF NO INTERRUPTS OCCURRED
1711	007112			ERRDF	26,,BVERR2,CKLOOP	;BEVENT CLAMP DID NOT PREVENT INTERRUPTS
1712	007112	104542		TRAP	T\$ERCODE	
1713	007114	000032		.WORD	26	
1714	007116	007212		.WORD	BVERR2	
1715	007120			4\$: CKLOOP		;CHECK FOR LOOP ON ERROR
1716	007120	104006		EMT	C\$CLP1	
1717	007122	005037	007142	CLR	ICOUNT	;CLEAR INTERRUPT COUNT
1718	007126			ENDSUB		
1719	007126			L10044:		
1720	007126	104003		EMT	C\$ESUB	
1721	007130			EXIT	TST	
1722	007130	104032		EMT	C\$EXIT	
1723	007132	000326		.WORD	L10042-	
1724	007134			INTSR:		
1725	007134			BGNSRV	BEVENT	;INTERRUPT SERVICE ROUTINE
1726	007134			BEVENT::		
1727	007134	005237	007142	INC	ICOUNT	;INCREMENT COUNTER
1728	007140			ENDSRV		
1729	007140			L10045:		
1730	007140	000002		RTI		
1731	007142			ICOUNT: .WORD	0	
1732	007142	000000				
1733	007144			BGNMSG	BVERR1	
1734	007144			BVERR1::		
1735	007144			PRINTB	#MSG1	
1736	007144			MOV	#MSG1,-(SP)	
1737	007144	012746	007260	MOV	#1,-(SP)	
1738	007150	012746	000001	MOV	SP,R0	
1739	007154	010600		EMT	C\$PNTB	
1740	007156	104014		ADD	#4,SP	
1741	007160	062706	000004	PRINTB	#INTCT,ICOUNT	
1742	007164			MOV	ICOUNT,-(SP)	
1743	007164	013746	007142	MOV	#INTCT,-(SP)	
1744	007170	012746	007334	MOV	#2,-(SP)	
1745	007174	012746	000002	MOV	SP,R0	
1746	007200	010600		EMT	C\$PNTB	
1747	007202	104014		ADD	#6,SP	
1748	007204	062706	000006	ENDMSG		
1749	007210			L10046:		
1750	007210			EMT	C\$MSG	
1751	007210	104023				
1752	007212			BGNMSG	BVERR2	
1753	007212			BVERR2::		
1754	007212			PRINTB	#MSG2	
1755	007212			MOV	#MSG2,-(SP)	
1756	007212	012746	007403	MOV	#1,-(SP)	
1757	007216	012746	000001	MOV	SP,R0	
1758	007222	010600		EMT	C\$PNTB	
1759	007224	104014				

1760	007226	062706	000004		ADD	#4,SP
1761	007232				PRINTB	#INTCT,ICOUNT
1762	007232	013746	007142		MOV	ICOUNT,-(SP)
1763	007236	012746	007334		MOV	#INTCT,-(SP)
1764	007242	012746	000002		MOV	#2,-(SP)
1765	007246	010600			MOV	SP,R0
1766	007250	104014			EMT	C\$PNTB
1767	007252	062706	000006		ADD	#6,SP
1768	007256				ENDMSG	
1769	007256			L10047:		
1770	007256	104023			EMT	C\$MSG
1771						
1772	007260	040445	042502	042526	MSG1:	.ASCIZ /%ABEVENT CLAMP FAILED TO ALLOW INTERRUPTS%N/
1773	007266	052116	041440	040514		
1774	007274	050115	043040	044501		
1775	007302	042514	020104	047524		
1776	007310	040440	046114	053517		
1777	007316	044440	052116	051105		
1778	007324	052522	052120	022523		
1779	007332	000116				
1780	007334	040445	052516	041115	INTCT:	.ASCIZ /%ANUMBER OF INTERRUPTS RECEIVED: %03%N/
1781	007342	051105	047440	020106		
1782	007350	047111	042524	051122		
1783	007356	050125	051524	051040		
1784	007364	041505	044505	042526		
1785	007372	035104	022440	031517		
1786	007400	047045	000			
1787	007403	045	041101	053105	MSG2:	.ASCIZ /%ABEVENT CLAMP DID NOT PREVENT INTERRUPTS%N/
1788	007410	047105	020124	046103		
1789	007416	046501	020120	044504		
1790	007424	020104	047516	020124		
1791	007432	051120	053105	047105		
1792	007440	020124	047111	042524		
1793	007446	051122	050125	051524		
1794	007454	047045	000			
1795		007460				
1796	007460				.EVEN	
1797	007460				ENDTST	
1798	007460	104001			L10042:	EMT C\$ETST

```

1799          .SBTTL TEST 4: LIGHT DISPLAY TEST
1800          :++
1801          :TEST TO VERIFY THAT THE FOUR RED LED'S ARE WORKING AND CAN BE
1802          :TURNED ON INDIVIDUALLY.
1803          :--
1804
1805 007462          BGNTST
1806
1807 007462 005037 177524          CLR      LSREG          ;TURN ON ALL FOUR LED'S
1808 007466          WAITMS #2.          ;DELAY APPROX. 0.2 SEC.
1809 007466 012700 000002          MOV      #2.,RO
1810 007472 104026          EMT      C$WTM
1811 007474 012737 000017 177524          MOV      #17,LSREG
1812 007502          WAITMS #2.          ;TURN OFF ALL FOUR LED'S
1813 007502 012700 000002          MOV      #2.,RO          ;DELAY APPROX. 0.2 SEC.
1814 007506 104026          EMT      C$WTM
1815 007510          MANUAL          ;IS MANUAL INTERVENTION ALLOWED?
1816 007510 104051          EMT      C$MANI
1817 007512          BCOMPLETE 2$          ;BR IF YES
1818 007512 103410          BCS      2$
1819 007514 022737 000030 002206          CMP      #30,PASCT          ;IS PASS COUNT >= 30?
1820 007522 003402          BLE      1$          ;BR IF YES
1821 007524          EXIT      TST
1822 007524 104032          EMT      C$EXIT
1823 007526 000072          .WORD   L10050-.
1824 007530 005037 002206 1$:          CLR      PASCT          ;EXIT TEST
1825 007534 012737 000016 177524 2$:          MOV      #16,LSREG          ;TURN ON THE LED CORRESPONDING TO THE LSB
1826 007542          WAITMS #2.          ;DELAY APPROX. 0.2 SEC.
1827 007542 012700 000002          MOV      #2.,RO
1828 007546 104026          EMT      C$WTM
1829 007550 012737 000015 177524          MOV      #15,LSREG
1830 007556          WAITMS #2.          ;TURN ON 2ND LED
1831 007556 012700 000002          MOV      #2.,RO          ;DELAY APPROX. 0.2 SEC.
1832 007562 104026          EMT      C$WTM
1833 007564 012737 000013 177524          MOV      #13,LSREG
1834 007572          WAITMS #2.          ;TURN ON 3RD LED
1835 007572 012700 000002          MOV      #2.,RO          ;DELAY APPROX. 0.2 SEC.
1836 007576 104026          EMT      C$WTM
1837 007600 012737 000007 177524          MOV      #7,LSREG
1838 007606          WAITMS #2.          ;TURN ON LED CORRESPONDING TO MSB
1839 007606 012700 000002          MOV      #2.,RO          ;DELAY APPROX. 0.2 SEC.
1840 007612 104026          EMT      C$WTM
1841 007614          EXIT      TST          ;EXIT
1842 007614 104032          EMT      C$EXIT
1843 007616 000002          .WORD   L10050-.
1844
1845 007620          ENDTST
1846 007620          L10050:
1847 007620 104001          EMT      C$ETST
1848
    
```



```

1849 .SBTTL TEST 5: ROCKER SWITCHES TEST
1850 :TEST TO CONFIRM THE ROCKER SWITCH SETTINGS. THIS TEST ASSUMES THAT,
1851 :IN MANUFACTURING, THE ROCKER SWITCHES ARE ALL IN THE ON POSITION.
1852 :THIS INCLUDES BOTH E21 AND E15. IN MANUFACTURING, THIS TEST WILL
1853 :VERIFY THAT ALL SWITCHES CAN BE READ AS ON. IN OTHER ENVIRONMENTS,
1854 :THE OPERATOR MAY SPECIFY WHAT THE SWITCH SETTINGS ARE BEFORE
1855 :THE DIAGNOSTIC IS STARTED (SEE PROGRAM OPTIONS UNDER OPERATING
1856 :INSTRUCTIONS). SWITCHES A1-A8 CORRESPOND TO E15 AND SWITCHES
1857 :B1-B4 TO E21.
1858 007622          BGNTST
1859
1860 007622          MANUAL          ;IS MANUAL INTERVENTION ALLOWED?
1861 007622 104051   EMT             C$MANI
1862 007624          BCOMPLETE      PRTSW          ;BR IF YES
1863 007624 103412   BCS             PRTSW
1864 007626 023737 002216 177524   CMP             SWSET,LSREG          ;ALL SWITCHES SHOULD BE ON AND BITS 0-11 SET
1865 007634 001403   BEQ             1$
1866 007636          ERRDF         27,SWERR          ;BR IF SWITCH READINGS ARE OK
1867 007636 104442   TRAP            T$ERCODE          ;CANNOT READ SWITCHES PROPERLY
1868 007640 000033   .WORD          27
1869 007642 010144   .WORD          SWERR
1870 007644          1$:          CKLOOP
1871 007644 104006   EMT             C$CLP1          ;CHECK FOR LOOP ON ERROR
1872 007646          EXIT
1873 007646 104032   EMT             TST
1874 007650 000524   .WORD          C$EXIT
1875 007652 013737 177524 010140 PRTSW:  L10051-
1876 007660 005037 010142          MOV             LSREG,TEMP          ;COPY CONTENTS OF LSREG
1877 007664 012737 000014 010136   CLR             SWCHON          ;CLEAR MASK
1878 007672 032737 000001 010140 LP:    MOV             #14,SWCNT          ;SET SWITCH COUNT
1879 007700 001403   BEQ             #BIT0,TEMP          ;TEST FOR SWITCH SET
1880 007702 052737 100000 010142   BIS             #BIT15,SWCHON      ;BR IF NOT SET
1881 007710 000241          CLC
1882 007712 006037 010142   ROR             #BIT15,SWCHON      ;IF SET, THEN SET CORRESPNDING BIT IN MASK
1883 007716 006037 010140   ROR             SWCHON            ;CLEAR C-BIT FOR ROTATE
1884 007722 005337 010136   ROR             TEMP              ;ROTATE SWSET
1885 007726 001361          DEC             SWCNT              ;GET READY TO TEST NEXT SWITCH
1886 007730 000241          BNE             LP                ;DECREMENT SWITCH COUNT
1887 007732 006037 010142   CLC
1888 007736 006037 010142   ROR             SWCHON            ;LOOP UNTIL ALL SWITCHES HAVE BEEN CHECKED
1889 007742 006037 010142   ROR             SWCHON            ;CLEAR C-BIT FOR ROTATE
1890 007746          PRINTF          #READN,SWCHON          ;ROTATE DATA
1891 007746 013746 010142   MOV             SWCHON,-(SP)        ;ROTATE DATA
1892 007752 012746 010176   MOV             #READN,-(SP)        ;ROTATE DATA
1893 007756 012746 000002   MOV             #2,-(SP)
1894 007762 010600          MOV             SP,R0
1895 007764 104017   EMT             C$PNTF
1896 007766 062706 000006   ADD             #6,SP
1897
1898 007772 013702 010142   MOV             SWCHON,R2          ;COPY SWITCH SETTINGS
1899 007776 012701 000001   MOV             #1,R1              ;SET SWITCH NUMBER = 1
1900 010002 032702 000001   TAG1:  BIT             #BIT0,R2          ;IS THIS SWITCH ON?
1901 010006 001411   BEQ             TAG2              ;BR IF NO
1902 010010          PRINTF          #MESSG1,R1          ;PRINT SWITCH NUMBER
1903 010010 010146   MOV             R1,-(SP)
1904 010012 012746 010227   MOV             #MESSG1,-(SP)
  
```

1905	010016	012746	000002		MOV	#2,-(SP)	
1906	010022	010600			MOV	SP,R0	
1907	010024	104017			EMT	C\$PNTF	
1908	010026	062706	000006		ADD	#6,SP	
1909	010032	005201		TAG2:	INC	R1	: INCREMENT SWITCH NUMBER
1910	010034	006002			ROR	R2	: ROTATE SWITCH REGISTER
1911	010036	022701	000010		CMP	#10,R1	: FINISHED WITH E15?
1912	010042	002357			BGE	TAG1	: BR IF NO
1913	010044	012701	000001		MOV	#1,R1	: RESET SWITCH NUMBER FOR E21
1914	010050	032702	000001	TAG3:	BIT	#BIT0,R2	: IS THIS SWITCH SET?
1915	010054	001411			BEQ	TAG4	: BR IF NO
1916	010056				PRINTF	#MESSG2,R1	: PRINT SWITCH NUMBER
1917	010056	010146			MOV	R1,-(SP)	
1918	010060	012746	010242		MOV	#MESSG2,-(SP)	
1919	010064	012746	000002		MOV	#2,-(SP)	
1920	010070	010600			MOV	SP,R0	
1921	010072	104017			EMT	C\$PNTF	
1922	010074	062706	000006		ADD	#6,SP	
1923	010100	005201		TAG4:	INC	R1	: INCREMENT SWITCH NUMBER
1924	010102	006002			ROR	R2	: ROTATE SWITCH REGISTER
1925	010104	022701	000004		CMP	#4,R1	: FINISHED?
1926	010110	002357			BGE	TAG3	: BR IF NO
1927	010112				PRINTF	#NEWLIN	
1928	010112	012746	010255		MOV	#NEWLIN,-(SP)	
1929	010116	012746	000001		MOV	#1,-(SP)	
1930	010122	010600			MOV	SP,R0	
1931	010124	104017			EMT	C\$PNTF	
1932	010126	062706	000004		ADD	#4,SP	
1933							
1934	010132				EXIT TST		
1935	010132	104032			EMT	C\$EXIT	
1936	010134	000240			.WORD	L10051-	
1937							
1938	010136	000000		SWCNT:	.WORD	0	
1939	010140	000000		TEMP:	.WORD	0	
1940	010142	000000		SWCHON:	.WORD	0	
1941							
1942	010144				BGNMSG	SWERR	
1943	010144			SWERR::			
1944	010144				PRINTB	#SERR1,SWSET,LSREG	
1945	010144	013746	177524		MOV	LSREG,-(SP)	
1946	010150	013746	002216		MOV	SWSET,-(SP)	
1947	010154	012746	010260		MOV	#SERR1,-(SP)	
1948	010160	012746	000003		MOV	#3,-(SP)	
1949	010164	010600			MOV	SP,R0	
1950	010166	104014			EMT	C\$PNTB	
1951	010170	062706	000010		ADD	#10,SP	
1952	010174				ENDMSG		
1953	010174			L10052:			
1954	010174	104023			EMT	C\$MSG	
1955							
1956	010176	040445	053523	052111	READN:	.ASCIZ	/%ASWITCHES ON : %06%A :
1957	010204	044103	051505	047440			
1958	010212	020116	020072	047445			
1959	010220	022466	020101	020072			
1960	010226	000					

1961	010227	045	040501	042045	MESSG1: .ASCIZ /%AA%D1%A, /
1962	010234	022461	026101	000040	
1963	010242	040445	022502	030504	MESSG2: .ASCIZ /%AB%D1%A, /
1964	010250	040445	020054	000	
1965	010255	045	000116		NEWLIN: .ASCIZ /%N/
1966	010260	040445	044504	020104	SERR1: .ASCII /%ADID NOT RECOGNIZE ALL SWITCHES AS ON%N/
1967	010266	047516	020124	042522	
1968	010274	047503	047107	055111	
1969	010302	020105	046101	020114	
1970	010310	053523	052111	044103	
1971	010316	051505	040440	020123	
1972	010324	047117	047045		
1973	010330	040445	054105	042520	.ASCIZ /%AEXPECTED: %06%S5%ARECEIVED:%06%N/
1974	010336	052103	042105	020072	
1975	010344	047445	022466	032523	
1976	010352	040445	042522	042503	
1977	010360	053111	042105	022472	
1978	010366	033117	047045	000	
1979		010374			
1980	010374				.EVEN
1981	010374				ENDTST
1982	010374	104001			L10051: EMT CBETST
1983					

```
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991 010376  
1992  
1993 010376  
1994 010376 104002  
1995 010400  
1996 010400 104051  
1997 010402  
1998 010402 103014  
1999 010404 005737 002204  
2000 010410 001032  
2001 010412  
2002 010412 104043  
2003 010414 000404  
2004 010416 002232  
2005 010420 000130  
2006 010422 011503  
2007 010424 000001  
2008 010426  
2009 010426 005737 002232  
2010 010432 001404  
2011 010434 012737 000400 010756  
2012 010442 000415  
2013 010444  
2014 010444 104043  
2015 010446 000406  
2016 010450 002220  
2017 010452 000042  
2018 010454 003004  
2019 010456 177777  
2020 010460 000000  
2021 010462 000024  
2022 010464  
2023 010464 004737 004742  
2024 010470 013737 002170 010756  
2025 010476 013737 010756 177520  
2026 010504 012737 000010 002172  
2027 010512 012705 002144  
2028 010516 012737 000001 002176  
2029 010524 005037 002164  
2030 010530 122737 177777 173774  
2031 010536 001005  
2032 010540  
2033 010540 104542  
2034 010542 000036  
2035 010544 010760  
2036 010546  
2037 010546 104032  
2038 010550 000754  
2039 010552
```

```
.SBTTL TEST 6: 2K DIAGNOSTIC ROM  
:++  
:TEST TO PERFORM CHECKSUM AND CHECKWORD VERIFICATION ON THE 2K  
:OF DIAGNOSTIC ROM. IN UNATTENDED MODE, THE ROM WILL BE ADDRESSED  
:FROM 0-2K. IN STAND-ALONE MODE, THE OPERATOR MAY CHANGE THE  
:ADDRESS BY RESPONDING TO QUESTIONS GENERATED ON THE FIRST PASS.  
:--  
BGNTST  
BGNSUB  
EMT C$BSUB  
MANUAL ;MANUAL INTERVENTION OK?  
EMT C$MANI  
BNCOMPLETE STRT ;BR IF NO  
BCC STRT  
TST PASS ;FIRST PASS?  
BNE RSTRT ;BR IF NO  
GMANIL DADDR,RSET,1,YES  
EMT C$GMAN  
BR 10000$  
.WORD RSET  
.WORD T$CODE  
.WORD DADDR  
.WORD 1  
10000$:  
TST RSET ;STANDARD JUMPERS?  
BEQ GETAD ;BR IF NO  
STRT: MOV #400,DRLP ;STORE STARTING ADDRESS  
BR RSTRT ;GO PERFORM TEST  
GETAD: GMANID LOADR,STORE,D,-1,0,24,NO  
EMT C$GMAN  
BR 10001$  
.WORD STORE  
.WORD T$CODE  
.WORD LOADR  
.WORD -1  
.WORD T$LOLIM  
.WORD T$HILIM  
10001$:  
JSR PC,SETADR ;GET STARTING ADDRESS  
MOV LOPAG,DRLP ;STORE STARTING ADDRESS  
RSTRT: MOV DRLP,PCR ;SET UP PCR  
DRTST: MOV #10,COUNTR ;SET NUMBER OF CHECKWORDS TO CHECK  
MOV #SFPTBL,R5 ;LOCATION OF CHECKWORDS  
MOV #1,RFLAG ;INDICATE ROM  
DLOOP: CLR BCF ;SIGNAL LOW BYTES ARE BEING CHECKED  
CMPB #-1,@#173774 ;DOES THE ROM EXIST?  
BNE 1$ ;BR IF YES  
ERRDF 30,DERR1,CKLOOP ;DIAGNOSTIC ROM E48 NOT FOUND  
TRAP T$ERRCODE  
.WORD 30  
.WORD DERR1  
EXIT TST ;EXIT TEST,ROM NOT FOUND  
EMT C$EXIT  
.WORD L:0053-  
1$: CKLOOP ;CHECK FOR LOOP ON ERROR
```


2040	010552	104006			EMT	C\$CLP1		
2041	010554	004737	004132		JSR	PC,CHKSUM	:	COMPUTE THE ACTUAL CHECKSUM
2042	010560	113737	173776	002200	MOVB	@#173776,EXPSUM	:	GET THE STORED CHECKSUM
2043	010566	063737	002202	002200	ADD	ACTSUM,EXPSUM	:	ADD THE EXPECTED AND ACTUAL CHECKSUMS
2044	010574	105737	002200		TSTB	EXPSUM	:	BYTE RESULT = 0?
2045	010600	001403			BEQ	2\$:	BR IF YES
2046	010602				ERRDF	31,,DERR2,CKLOOP	:	CHECKSUM ERROR IN DIAGNOSTIC ROM
2047	010602	104542			TRAP	T\$ERCODE		
2048	010604	000037			.WORD	31		
2049	010606	011006			.WORD	DERR2		
2050	010610				2\$:	CKLOOP	:	CHECK FOR LOOP ON ERROR
2051	010610	104006			EMT	C\$CLP1		
2052	010612				ENDSUB			
2053	010612				L10054:			
2054	010612	104003			EMT	C\$ESUB		
2055								
2056								
2057	010614				BGNSUB			
2058	010614	104002			EMT	C\$BSUB		
2059	010616	012737	000001	00?164	MOV	#1,BCF	:	SET BCF TO DENOTE HIGH BYTES
2060	010624	122737	177777	175775	CMPB	#-1,@#173775	:	DOES THE ROM EXIST?
2061	010632	001005			BNE	3\$:	BR IF YES
2062	010634				ERRDF	32,,DERR3,CKLOOP	:	DIAGNOSTIC ROM E53 NOT FOUND
2063	010634	104542			TRAP	T\$ERCODE		
2064	010636	000040			.WORD	32		
2065	010640	011034			.WORD	DERR3		
2066	010642				EXIT	TST	:	EXIT TEST, ROM NOT FOUND
2067	010642	104032			EMT	C\$EXIT		
2068	010644	000660			.WORD	L10053-		
2069	010646				3\$:	CKLOOP	:	CHECK FOR LOOP ON ERROR
2070	010646	104006			EMT	C\$CLP1		
2071	010650	004737	004132		JSR	PC,CHKSUM	:	COMPUTE THE ACTUAL CHECKSUM
2072	010654	113737	173777	002200	MOVB	@#173777,EXPSUM	:	GET EXPECTED CHECKSUM
2073	010662	063737	002202	002200	ADD	ACTSUM,EXPSUM	:	ADD THE EXPECTED AND ACTUAL CHECKSUMS
2074	010670	105737	002200		TSTB	EXPSUM	:	BYTE RESULT = 0?
2075	010674	001403			BEQ	4\$:	BR IF YES
2076	010676				ERRDF	33,,DERR4,CKLOOP	:	CHECKSUM ERROR IN DIAGNOSTIC ROM
2077	010676	104542			TRAP	T\$ERCODE		
2078	010700	000041			.WORD	33		
2079	010702	011062			.WORD	DERR4		
2080	010704				4\$:	CKLOOP	:	CHECK FOR LOOP ON ERROR
2081	010704	104006			EMT	C\$CLP1		
2082	010706				ENDSUB			
2083	010706				L10055:			
2084	010706	104003			EMT	C\$ESUB		
2085								
2086	010710				BGNSUB			
2087	010710	104002			EMT	C\$BSUB		
2088	010712	022537	173776		CMP	(R5)+,@#173776	:	VERIFY THE CHECKWORD FOR THIS PAGE
2089	010716	001403			BEQ	5\$:	BR IF THE SAME
2090	010720				ERRDF	34,,DERR5,CKLOOP	:	CHECKWORD ERROR
2091	010720	104542			TRAP	T\$ERCODE		
2092	010722	000042			.WORD	34		
2093	010724	011110			.WORD	DERR5		
2094	010726				5\$:	CKLOOP	:	CHECK FOR LOOP ON ERROR
2095	010726	104006			EMT	C\$CLP1		

2096	010730				ENDSUB
2097	010730			L10056:	
2098	010730	104003			EMT C\$ESUB
2099					
2100					
2101	010732				BGNSUB
2102	010732	104002			EMT C\$BSUB
2103	010734	062737	001002	177520	ADD #1002,PCR
2104	010742	005337	002172		DEC COUNTR
2105	010746	001266			BNE DLOOP
2106	010750				ENDSUB
2107	010750			L10057:	
2108	010750	104003			EMT C\$ESUB
2109					
2110	010752				EXIT TST
2111	010752	104032			EMT C\$EXIT
2112	010754	000550			.WORD L10053-
2113					
2114	010756	000000			DRLP: .WORD 0
2115					
2116	010760			BGNMSG	DERR1
2117	010760			DERR1::	
2118	010760			PRINTB	#LRAERR,#NODR
2119	010760	012746	011457		MOV #NODR,-(SP)
2120	010764	012746	011136		MOV #LRAERR,-(SP)
2121	010770	012746	000002		MOV #2,-(SP)
2122	010774	010600			MOV SP,R0
2123	010776	104014			EMT C\$PNTB
2124	011000	062706	000006		ADD #6,SP
2125	011004			ENDMSG	
2126	011004			L10060:	
2127	011004	104023			EMT C\$MSG
2128					
2129	011006			BGNMSG	DERR2
2130	011006			DERR2::	
2131	011006			PRINTB	#LOWROM,#CKERR
2132	011006	012746	002634		MOV #CKERR,-(SP)
2133	011012	012746	011212		MOV #LOWROM,-(SP)
2134	011016	012746	000002		MOV #2,-(SP)
2135	011022	010600			MOV SP,R0
2136	011024	104014			EMT C\$PNTB
2137	011026	062706	000006		ADD #6,SP
2138	011032			ENDMSG	
2139	011032			L10061:	
2140	011032	104023			EMT C\$MSG
2141					
2142	011034			BGNMSG	DERR3
2143	011034			DERR3::	
2144	011034			PRINTB	#HRAERR,#NODR
2145	011034	012746	011457		MOV #NODR,-(SP)
2146	011040	012746	011261		MOV #HRAERR,-(SP)
2147	011044	012746	000002		MOV #2,-(SP)
2148	011050	010600			MOV SP,R0
2149	011052	104014			EMT C\$PNTB
2150	011054	062706	000006		ADD #6,SP
2151	011060			ENDMSG	

:NEXT PAGE IN PCR
 :DECREMENT CHECKWORD COUNT
 :LOOP UNTIL ALL 20 PAGES HAVE BEEN CHECKED

2152	011060				L10062:	
2153	011060	104023			EMT	C\$MSG
2154						
2155	011062				BGNMSG	DERR4
2156	011062				DERR4::	
2157	011062				PRINTB	#HIROM,#CKERR
2158	011062	012746	002634		MOV	#CKERR,-(SP)
2159	011066	012746	011335		MOV	#HIROM,-(SP)
2160	011072	012746	000002		MOV	#2,-(SP)
2161	011076	010600			MOV	SP,R0
2162	011100	104014			EMT	C\$PNTB
2163	011102	062706	000006		ADD	#6,SP
2164	011106				ENDMSG	
2165	011106				L10063:	
2166	011106	104023			EMT	C\$MSG
2167						
2168	011110				BGNMSG	DERR5
2169	011110				DERR5::	
2170	011110				PRINTB	#MISTAK
2171	011110	012746	011405		MOV	#MISTAK,-(SP)
2172	011114	012746	000001		MOV	#1,-(SP)
2173	011120	010600			MOV	SP,R0
2174	011122	104014			EMT	C\$PNTB
2175	011124	062706	000004		ADD	#4,SP
2176	011130	004737	004356		JSR	PC,VIRTAD
2177	011134				ENDMSG	
2178	011134				L10064:	
2179	011134	104023			EMT	C\$MSG
2180						
2181	011136	052045	047045	040445	LRAERR:	.ASCIZ /%T%N%ACANNOT ACCESS DIAGNOSTIC ROM IN E48%N/
2182	011144	040503	047116	052117		
2183	011152	040440	041503	051505		
2184	011160	020123	044504	043501		
2185	011166	047516	052123	041511		
2186	011174	051040	046517	044440		
2187	011202	020116	032105	022470		
2188	011210	000116				
2189						
2190	011212	052045	047045	040445	LOWROM:	.ASCIZ /%T%N%ALOW BYTE DIAGNOSTIC ROM IN E48%N/
2191	011220	047514	020127	054502		
2192	011226	042524	042040	040511		
2193	011234	047107	051517	044524		
2194	011242	020103	047522	020115		
2195	011250	047111	042440	034064		
2196	011256	047045	000			
2197						
2198	011261	045	022524	022516	HRAERR:	.ASCIZ /%T%N%ACANNOT ACCESS DIAGNOSTIC ROM IN E53%N/
2199	011266	041501	047101	047516		
2200	011274	020124	041501	042503		
2201	011302	051523	042040	040511		
2202	011310	047107	051517	044524		
2203	011316	020103	047522	020115		
2204	011324	047111	042440	031465		
2205	011332	047045	000			
2206						
2207	011335	045	022524	022516	HIROM:	.ASCIZ /%T%N%AHIGH BYTE DIAGNOSTIC ROM IN E53%N/

2208	011342	044101	043511	020110	
2209	011350	054502	042524	042040	
2210	011356	040511	047107	051517	
2211	011364	044524	020103	047522	
2212	011372	020115	047111	042440	
2213	011400	031465	047045	000	
2214					
2215	011405	045	044501	041516	MISTAK: .ASCIZ /%AINCORRECT CHECKWORD IN DIAGNOSTIC ROM%/
2216	011412	051117	042522	052103	
2217	011420	041440	042510	045503	
2218	011426	047527	042122	044440	
2219	011434	020116	044504	043501	
2220	011442	047516	052123	041511	
2221	011450	051040	046517	047045	
2222	011456	000			
2223					
2224	011457	116	047117	042455	NODR: .ASCIZ /NON-EXISTENT MEMORY/
2225	011464	044530	052123	047105	
2226	011472	020124	042515	047515	
2227	011500	054522	000		
2228					
2229	011503	123	040524	042116	DADDR: .ASCIZ /STANDARD JUMPERS/
2230	011510	051101	020104	052512	
2231	011516	050115	051105	000123	
2232					
2233					
2234	011524				.EVEN ENDTST
2235	011524				
2236	011524	10400!			L10053: EMT CSETST

2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292

011526
011526 104002
011530 104051
011532 103112
011534 005037 013540
011540 005737 002204
011544 001422
011546 005737 013544
011552 001153
011554 005737 013546
011560 001402
011562 000137 012430
011566 005737 013550
011572 001402
011574 000137 012776
011600 005737 013552
011604 001465
011606 000137 013334
011612
011612 104043
011614 000404
011616 013540
011620 000130
011622 014047
011624 000001
011626
011626 005737 013540
011632 001452

```
.SBTTL TEST 7: TEST ALL ADDITIONAL MEMORY
:++
:TEST TO LOCATE AND VERIFY CHECKSUMS IN ALL RESIDENT MEMORY
:ON A PAGE BASIS. THERE ARE FOUR STORAGE AREAS ASSOCIATED
:WITH THIS TEST WHICH HOLD THE CHECKWORDS OF ALL THE MEMORY
:THAT IS TO BE TESTED. THESE TABLES WILL HAVE DEFAULT VALUES
:ONLY IF THE ASSOCIATED MEMORY CHIP IS A STANDARD COMPONENT
:ON THE BOARD. IF NO DEFAULT VALUES EXIST, THE OPERATOR MUST
:INPUT THE CHECKWORDS AS LISTED ON THE PRINT SET. THE MEMORY
:WILL BE TESTED IN THE FOLLOWING LOCATIONS BY DEFAULT:
:
:   EXPANDABLE DIAGNOSTIC ROM      2-4K
:   EPROM IN SOCKETS              4-6K
:   SYSTEM ROM                    16-32K
:   SYSTEM EPROM                  16-24K
:
:THE TEST WILL FIRST VERIFY THE CHECKSUMS IN ALL RESIDENT ROM,
:THEN COMPARE THE ACTUAL CHECKWORDS. ERROR INFORMATION WILL
:INCLUDE THE SPECIFIC TYPE OF ERROR THAT OCCURRS, THE VIRTUAL
:ADDRESS, AND WHETHER IT WAS THE HIGH BYTE OR LOW BYTE ROM/EPROM.
:THIS INFORMATION SHOULD ALLOW A KNOWLEDGEABLE OPERATOR TO ISOLATE
:THE ERROR DOWN TO A SINGLE ROM/EPROM WITH THE AID OF THE
:ADDRESS MAP IN THE PRINT SET.
:--
      BGNTST
      BGNSUB
      EMT      ($BSUB
      MANUAL
      EMT      ($MANI
      BNCOMPLET     DFLTST
      BCC      DFLTST
      CLR      ADDON
      TST      PASS
      BEQ      GET
      TST      LOD1
      BNE      LD1
      TST      LOD2
      BEQ      P1
      JMP      LD2
      P1:     TST      LOD3
      BEQ      P2
      JMP      LD3
      P2:     TST      LOD4
      BEQ      DFLTST
      JMP      LD4
      GET:    GMANIL   EXEC,ADDON,1,YES
      EMT      ($GMAN
      BR       10000$
      .WORD   ADDON
      .WORD   T$CODE
      .WORD   EXEC
      .WORD   1
      10000$: TST      ADDON
      BEQ      DFLTST
:UNDER APT?
:SKIP TEST IF YES
:RESTORE DEFAULT
:FIRST PASS?
:BR IF YES
:EXPANDED DIAGNOSTIC ROM?
:BR IF YES
:EPROM IN SOCKETS?
:BR IF NO
:TEST EPROM
:SYSTEM ROM ?
:BR IF NO
:TEST ROM
:SYSTEM EPROM?
:EXIT IF NO
:TEST EPROM
:ADDITIONAL MEMORY?
:BR IF NO
```

2293	011634			DIAIN:	GMANIL	EXPND,RESPND,1,NO	
2294	011634	104043			EMT	C\$GMAN	
2295	011636	000404			BR	10001\$	
2296	011640	002230			.WORD	RESPND	
2297	011642	000120			.WORD	T\$CODE	
2298	011644	014076			.WORD	EXPND	
2299	011646	000001			.WORD	1	
2300	011650			10001\$:			
2301	011650	005737	002230		TST	RESPND	:EXPANDED DIAGNOSTIC ROM?
2302	011654	001045			BNE	EXPROM	:BR IF YES
2303	011656			EPRIN:	GMANIL	EPRM,RESPND,1,NO	
2304	011656	104043			EMT	C\$GMAN	
2305	011660	000404			BR	10002\$	
2306	011662	002230			.WORD	RESPND	
2307	011664	000120			.WORD	T\$CODE	
2308	011666	014126			.WORD	EPRM	
2309	011670	000001			.WORD	1	
2310	011672			10002\$:			
2311	011672	005737	002230		TST	RESPND	:EPROM IN SOCKETS?
2312	011676	001402			BEQ	SYSRIN	:BR IF NO
2313	011700	000137	012316		JMP	EPRMT	:JUMP TO ACCEPT INPUT
2314	011704			SYSRIN:	GMANIL	SYSR,RESPND,1,NO	
2315	011704	104043			EMT	C\$GMAN	
2316	011706	000404			BR	10003\$	
2317	011710	002230			.WORD	RESPND	
2318	011712	000120			.WORD	T\$CODE	
2319	011714	014147			.WORD	SYSR	
2320	011716	000001			.WORD	1	
2321	011720			10003\$:			
2322	011720	005737	002230		TST	RESPND	:SYSTEM ROM?
2323	011724	001402			BEQ	SYSEIN	:BR IF NO
2324	011726	000137	012636		JMP	SYSRT	:INPUT CHECKWORDS
2325	011732			SYSEIN:	GMANIL	SYSE,RESPND,1,NO	
2326	011732	104043			EMT	C\$GMAN	
2327	011734	000404			BR	10004\$	
2328	011736	002230			.WORD	RESPND	
2329	011740	000120			.WORD	T\$CODE	
2330	011742	014162			.WORD	SYSE	
2331	011744	000001			.WORD	1	
2332	011746			10004\$:			
2333	011746	005737	002230		TST	RESPND	:SYSTEM EPROM?
2334	011752	001402			BEQ	DFLTST	:BR IF NO
2335	011754	000137	013174		JMP	SYSET	:INPUT CHECKWORDS
2336	011760			DFLTST:	EXIT	TST	:NO ADDTL. MEMORY -- EXIT
2337	011760	104032			EMT	C\$EXIT	
2338	011762	002344			.WORD	L10065-	
2339	011764			L10066:	ENDSUB		
2340	011764						
2341	011764	104003			EMT	C\$ESUB	
2342							
2343							
2344	011766				BGNSUB		
2345	011766	104002			EMT	C\$BSUB	
2346	011770	005037	002242	EXPROM:	CLR	ERRFLG	:CLEAR ERROR FLAG
2347	011774	012737	000010		MOV	#10,WORDCT	:COUNT 8 CHECKWORDS
2348	012002	012702	002244	002222	MOV	#EXPDIA,R2	:POINTER TO STORAGE TABLE

2349	012006	004737	004176			JSR	PC,INPUT		;INPUT CHECKWORDS
2350	012012					GMANIL	EXADD,ANSR,1,YES		
2351	012012	104043				EMT	C\$GMAN		
2352	012014	000404				BR	10000\$		
2353	012016	002174				.WORD	ANSR		
2354	012020	000130				.WORD	T\$CODE		
2355	012022	014177				.WORD	EXADD		
2356	012024	000001				.WORD	1		
2357	012026								
2358	012026	005737	002174			TST	ANSR		;STANDARD MEMORY RANGE?
2359	012032	001020				BNE	1\$;BR IF YES
2360	012034	005237	002174			INC	ANSR		;RESTORE DEFAULT VALUE
2361	012040					GMANID	LOADR,STORE,D,-1,0,30,NO		
2362	012040	104043				EMT	C\$GMAN		
2363	012042	000406				BR	10001\$		
2364	012044	002220				.WORD	STORE		
2365	012046	000042				.WORD	T\$CODE		
2366	012050	003004				.WORD	LOADR		
2367	012052	177777				.WORD	-1		
2368	012054	000000				.WORD	T\$LOLIM		
2369	012056	000030				.WORD	T\$HILIM		
2370	012060								
2371	012060	004737	004742			JSR	PC,SETADR		;GET FIRST PAGE ADDRESS
2372	012064	013737	002170	013544		MOV	LOPAG,LOD1		;STORE LOW PAGE NO.
2373	012072	000403				BR	LD1		;SKIP NEXT INSTRUCTION
2374	012074	012737	010420	013544	1\$:	MOV	#010420,LOD1		;STANDARD PAGE = 20,21 2-4K RANGE
2375	012102	013737	013544	177520	LD1:	MOV	LOD1,PCR		;LOAD STARTING PAGE
2376	012110	012737	000001	002176		MOV	#1,RFLAG		;INDICATE ROM
2377	012116	012703	002244			MOV	#EXPDIA,R3		;POINTER TO CHECKWORDS
2378	012122	012737	000010	002172		MOV	#10,COUNTR		;PAGE COUNT
2379	012130	012337	002226		EXPTST:	MOV	(R3)+,CKWD		;GET CHECKWORD FOR THIS PAGE
2380	012134	004737	004514			JSR	PC,MEMTST		;TEST MEMORY
2381	012140	005737	002166			TST	REAL		;DOES THE MEMORY EXIST?
2382	012144	001455				BEQ	E3		;BR IF NO
2383	012146	005737	002242			TST	ERRFLG		;ANY OTHER ERRORS?
2384	012152	001421				BEQ	NOERR		;BR IF NO
2385	012154	004737	004356			JSR	PC,VIRTAD		;GET ADDRESS OF ERROR
2386	012160	005737	002164			TST	BCF		;LOW BYTE PAGE?
2387	012164	001004				BNE	HIGH		;BR IF NO
2388	012166	012737	002677	002240		MOV	#LOBYT,BYTLOC		;SET POINTER FOR ERROR MSG.
2389	012174	000403				BR	DATOUT		;PRINT ERROR MESSAGE
2390	012176	012737	002741	002240	HIGH:	MOV	#HIBYT,BYTLOC		;POINTER FOR ERROR MSG.
2391	012204	022737	000001	002242	DATOUT:	CMP	#1,ERRFLG		;CHECKSUM ERROR?
2392	012212	001420				BEQ	E1		;BR IF YES
2393	012214	000424				BR	E2		;ELSE CHECKWORD ERROR
2394	012216	062737	001002	177520	NOERR:	ADD	#1002,PCR		;ADJUST PCR
2395	012224	005337	002172			DEC	COUNTR		;DEC PAGE COUNT
2396	012230	001337				BNE	EXPTST		;LOOP UNTIL ALL PAGES ARE TESTED
2397	012232	005737	002204		MORE:	TST	PASS		;FIRST PASS?
2398	012236	001002				BNE	1\$;BR IF NO
2399	012240	000137	011656			JMP	EPRIN		;TEST ANY ADDITIONAL MEMORY
2400	012244	000137	011566		1\$:	JMP	P1		;FIND ANY ADDITIONAL MEMORY
2401	012250					EXIT	SUB		;END OF SUBTEST
2402	012250	104032				EMT	C\$EXIT		
2403	012252	000040				.WORD	L:0067-		
2404	012254				E1:	ERRDF	35,EXPND,CKSME,CKLOOP		

2405	012254	104562				TRAP	T\$ERCODE	
2406	012256	000043				.WORD	35	
2407	012260	014076				.WORD	EXPND	
2408	012262	013554				.WORD	CKSME	
2409	012264	000762				BR	MORE	
2410	012266			E2:		ERRDF	36,EXPND,CWKDE,CKLOOP	
2411	012266	104562				TRAP	T\$ERCODE	
2412	012270	000044				.WORD	36	
2413	012272	014076				.WORD	EXPND	
2414	012274	013612				.WORD	CWKDE	
2415	012276	000755				BR	MORE	
2416	012300			E3:		ERRDF	37,EXPND,NONXT,CKLOOP	
2417	012300	104562				TRAP	T\$ERCODE	
2418	012302	000045				.WORD	37	
2419	012304	014076				.WORD	EXPND	
2420	012306	013644				.WORD	NONXT	
2421	012310	000750				BR	MORE	
2422	012312					ENDSUB		
2423	012312			L10067:				
2424	012312	104003				EMT	C\$ESUB	
2425								
2426	012314					BGNSUB		
2427	012314	104002				EMT	C\$BSUB	
2428	012316	005037	002242		EPRMT:	CLR	ERRFLG	:CLEAR ERROR FLAG
2429	012322	012737	000010	002222		MOV	#10,WORDCT	:INPUT 8 CHECKWORDS
2430	012330	012702	002264			MOV	#EPROM,R2	:POINTER TO STORAGE TABLE
2431	012334	004737	004176			JSR	PC,INPUT	:INPUT CHECKWORDS
2432	012340					GMANIL	EPADD,ANSR,1,YES	
2433	012340	104043				EMT	C\$GMAN	
2434	012342	000404				BR	10000\$	
2435	012344	002174				.WORD	ANSR	
2436	012346	000130				.WORD	T\$CODE	
2437	012350	014225				.WORD	EPADD	
2438	012352	000001				.WORD	1	
2439	012354			10000\$:				
2440	012354	005737	002174			TST	ANSR	:STANDARD MEMORY RANGE?
2441	012360	001020				BNE	1\$:BR IF YES
2442	012362	005237	002174			INC	ANSR	:RESTORE DEFAULT
2443	012366					GMANID	LOADR,STORE,D,-1,0,30,NO	
2444	012366	104043				EMT	C\$GMAN	
2445	012370	000406				BR	10001\$	
2446	012372	002220				.WORD	STORE	
2447	012374	000042				.WORD	T\$CODE	
2448	012376	003004				.WORD	LOADR	
2449	012400	177777				.WORD	-1	
2450	012402	000000				.WORD	T\$LOLIM	
2451	012404	000030				.WORD	T\$HILIM	
2452	012406			10001\$:				
2453	012406	004737	004742			JSR	PC,SETADR	:GET FIRST PAGE ADDRESS
2454	012412	013737	002170	013546		MOV	LOPAG,LOD2	:STORE LOW PAGE NO.
2455	012420	000403				BR	LD2	:SKIP NEXT INSTRUCTION
2456	012422	012737	020440	013546	1\$:	MOV	#020440,LOD2	:STANDARD PAGE = 40,41 4-6K RANGE
2457	012430	013737	013546	177520	LD2:	MOV	LOD2,PCR	:LOAD STARTING ADDRESS
2458	012436	005037	002176			CLR	RFLAG	:INDICATE EPROM
2459	012442	012703	002264			MOV	#EPROM,R3	:POINT TO CHECKWORDS
2460	012446	012737	000010	002172		MOV	#10,COUNTR	:PAGE COUNT

2461	012454	012337	002226		EPRTST: MOV	(R3)+,CKWD	:GET CHECKWORD FOR THIS PAGE
2462	012460	004737	004514		JSR	PC,MEMTST	:TEST MEMORY
2463	012464	005737	002166		TST	REAL	:DOES THE MEMORY EXIST?
2464	012470	001453			BEQ	E6	:BR IF NO
2465	012472	005737	002242		TST	ERRFLG	:ANY OTHER ERRORS?
2466	012476	001421			BEQ	NONE	:BR IF NO
2467	012500	004737	004356		JSR	PC,VIRTAD	:GET ADDRESS OF ERROR
2468	012504	005737	002164		TST	BCF	:LOW BYTE PAGE?
2469	012510	001004			BNE	HIADD	:BR IF NO
2470	012512	012737	002677	002240	MOV	#LOBYT,BYTLOC	:SET POINTER FOR ERROR MSG.
2471	012520	000403			BR	PRIOUT	:PRINT ERROR MESSAGE
2472	012522	012737	002741	002240	HIADD: MOV	#HIBYT,BYTLOC	:POINTER FOR ERROR MSG.
2473	012530	022737	000001	002242	PRIOUT: CMP	#1,ERRFLG	:CHECKSUM ERROR?
2474	012536	001416			BEQ	E4	:BR IF YES
2475	012540	000422			BR	E5	:ELSE CHECKWORD ERROR
2476	012542	062737	001002	177520	NONE: ADD	#1002,PCR	:ADJUST PAGE IN PCR
2477	012550	005337	002172		DEC	COUNTR	:DEC PAGE COUNT
2478	012554	001337			BNE	EPRTST	:LOOP UNTIL FINISHED
2479	012556	005737	002204		ADDTL: TST	PASS	:FIRST PASS?
2480	012562	001002			BNE	1\$:BR IF NO
2481	012564	000137	011704		JMP	SYSRIN	:TEST ANY ADDITIONAL MEMORY
2482	012570	000137	011600		1\$: JMP	P2	:FIND ANY ADDITIONAL MEMORY
2483	012574				E4: ERRDF	40,EPRM,CKSME,CKLOOP	
2484	012574	104562			TRAP	T\$ERCODE	
2485	012576	000050			.WORD	40	
2486	012600	014126			.WORD	EPRM	
2487	012602	013554			.WORD	CKSME	
2488	012604	000764			BR	ADDTL	
2489	012606				E5: ERRDF	41,EPRM,CWKDE,CKLOOP	
2490	012606	104562			TRAP	T\$ERCODE	
2491	012610	000051			.WORD	41	
2492	012612	014126			.WORD	EPRM	
2493	012614	013612			.WORD	CWKDE	
2494	012616	000757			BR	ADDTL	
2495	012620				E6: ERRDF	42,EPRM,NONXT,CKLOOP	
2496	012620	104562			TRAP	T\$ERCODE	
2497	012622	000052			.WORD	42	
2498	012624	014126			.WORD	EPRM	
2499	012626	013644			.WORD	NONXT	
2500	012630	000752			BR	ADDTL	
2501	012632				ENDSUB		
2502	012632				L10070:		
2503	012632	104003			EMT	C\$ESUB	
2504							
2505	012634				BGNSUB		
2506	012634	104002			EMT	C\$BSUB	
2507	012636	005037	002242		CLR	ERRFLG	:CLEAR ERROR FLAG
2508	012642				GMANID	RWDCT,RESPND,D,-1,10,100,NO	
2509	012642	104043			EMT	C\$GMAN	
2510	012644	000406			BR	10000\$	
2511	012646	002230			.WORD	RESPND	
2512	012650	000042			.WORD	T\$CODE	
2513	012652	014005			.WORD	RWDCT	
2514	012654	177777			.WORD	-1	
2515	012656	000010			.WORD	T\$OLIM	
2516	012660	000100			.WORD	T\$HILIM	

```

2517 012662          10000$:
2518 012662 013737 002230 013542 MOV     RESPND,PGCT      ;STORE PAGE COUNT
2519 012670 013737 002230 002222 MOV     RESPND,WORDCT   ;COPY WORD COUNT
2520 012676 012702 002304          MOV     #SYSROM,R2     ;POINTER TO STORAGE TABLE
2521 012702 004737 004176          JSR     PC,INPUT       ;INPUT CHECKWORDS
2522 012706          GMANIL  SRR,ANSR,1,YES
2523 012706 104043          EMT     C$GMAN
2524 012710 000404          BR      10001$
2525 012712 002174          .WORD  ANSR
2526 012714 000130          .WORD  T$CODE
2527 012716 014244          .WORD  SRR
2528 012720 000001          .WORD  1
2529 012722          10001$:
2530 012722 005737 002174          TST     ANSR           ;STANDARD MEMORY RANGE?
2531 012726 001020          BNE     1$            ;BR IF YES
2532 012730 005237 002174          INC     ANSR          ;RESTORE DEFAULT VALUE
2533 012734          GMANID  LOADR,STORE,D,-1,0,30,NO
2534 012734 104043          EMT     C$GMAN
2535 012736 000406          BR      10002$
2536 012740 002220          .WORD  STORE
2537 012742 000042          .WORD  T$CODE
2538 012744 003004          .WORD  LOADR
2539 012746 177777          .WORD  -1
2540 012750 000000          .WORD  T$LOLIM
2541 012752 000030          .WORD  T$HILIM
2542 012754          10002$:
2543 012754 004737 004742          JSR     PC,SETADR     ;GET FIRST PAGE ADDRESS
2544 012760 013737 002170 013550 MOV     LOPAG,LOD3     ;STORE LOW PAGE NO.
2545 012766 000403          BR      LD3          ;SKIP NEXT INSTRUCTION
2546 012770 012737 100600 013550 1$: MOV     #100600,LOD3   ;STANDARD PAGE = 200,201 16-32K RANGE
2547 012776 013737 013550 177520 LD3: MOV     LOD3,PCR   ;LOAD STARTING ADDRESS
2548 013004 012737 000001 002176 MOV     #1,RFLAG      ;INDICATE ROM
2549 013012 012703 002304          MOV     #SYSROM,R3   ;POINT TO CHECKWORDS
2550 013016 013737 013542 002172 MOV     PGCT,COUNTR   ;PAGE COUNT
2551 013024 012337 002226          SYRTST: MOV    (R3)+,CKWD ;GET CHECKWORD FOR THIS PAGE
2552 013030 004737 004514          JSR     PC,MEMTST    ;TEST MEMORY
2553 013034 005737 002166          TST     REAL         ;DOES THE MEMORY EXIST?
2554 013040 001446          BEQ     E11         ;BR IF NO
2555 013042 005737 002242          TST     ERRFLG      ;ANY OTHER ERRORS?
2556 013046 001421          BEQ     PASSED      ;BR IF NO
2557 013050 004737 004356          JSR     PC,VIRTAD    ;GET ADDRESS OF ERROR
2558 013054 005737 002164          TST     BCF         ;LOW BYTE PAGE?
2559 013060 001004          BNE     HIGHB       ;BR IF NO
2560 013062 012737 002677 002240 MOV     #LOBYT,BYTLOC ;SET POINTER FOR ERROR MSG.
2561 013070 000403          BR      MSGOUT      ;PRINT ERROR MESSAGE
2562 013072 012737 002741 002240 HIGHB: MOV    #HIBYT,BYTLOC ;POINTER FOR ERROR MSG.
2563 013100 022737 000001 002242 MSGOUT: CMP     #1,ERRFLG  ;CHECKSUM ERROR?
2564 013106 001411          BEQ     E7          ;BR IF YES
2565 013110 000415          BR      E10         ;ELSE CHECKWORD ERROR
2566 013112 062737 001002 177520 PASSED: ADD    #1002,PCR  ;ADJUST PAGE IN PCR
2567 013120 005337 002172          DEC     COUNTR      ;DEC PAGE COUNT
2568 013124 001337          BNE     SYRTST      ;LOOP UNTIL FINISHED
2569 013126          NEXT:  EXIT     TST     ;TEST IS FINISHED
2570 013126 104032          EMT     C$EXIT
2571 013130 001176          .WORD  L:0065-
2572 013132          E7:  ERRDF  43,SYSR,CKSME,CKLOOP
    
```


2573	013132	104562			TRAP	T\$ERCODE	
2574	013134	000053			.WORD	43	
2575	013136	014147			.WORD	SYSR	
2576	013140	013554			.WORD	CKSME	
2577	013142	000771			BR	NEXT	
2578	013144		E10:		ERRDF	44,SYSR,CWKDE,CKLOOP	
2579	013144	104562			TRAP	T\$ERCODE	
2580	013146	000054			.WORD	44	
2581	013150	014147			.WORD	SYSR	
2582	013152	013612			.WORD	CWKDE	
2583	013154	000764			BR	NEXT	
2584	013156		E11:		ERRDF	45,SYSR,NONXT,CKLOOP	
2585	013156	104562			TRAP	T\$ERCODE	
2586	013160	000055			.WORD	45	
2587	013162	014147			.WORD	SYSR	
2588	013164	013644			.WORD	NONXT	
2589	013166	000757			BR	NEXT	
2590	013170				ENDSUB		
2591	013170		L10071:				
2592	013170	104003			EMT	C\$ESUB	
2593							
2594	013172				BGNSUB		
2595	013172	104002			EMT	C\$BSUB	
2596	013174	005037	002242		CLR	ERRFLG	:CLEAR ERROR FLAG
2597	013200				G\$MANID	RWDCT,RESPND,D,-1,10,40,NO	
2598	013200	104043			EMT	C\$G\$MAN	
2599	013202	000406			BR	10000\$	
2600	013204	002230			.WORD	RESPND	
2601	013206	000042			.WORD	T\$CODE	
2602	013210	014005			.WORD	RWDCT	
2603	013212	177777			.WORD	-1	
2604	013214	000010			.WORD	T\$LOLIM	
2605	013216	000040			.WORD	T\$HILIM	
2606	013220						
2607	013220	013737	002230	013542	MOV	RESPND,PGCT	:STORE PAGE COUNT
2608	013226	013737	002230	002222	MOV	RESPND,WORDCT	:COPY WORD COUNT
2609	013234	012702	002304		MOV	#SYSROM,R2	:POINTER TO STORAGE TABLE
2610	013240	004737	004176		JSR	PC,INPUT	:INPUT CHECKWORDS
2611	013244				G\$MANIL	SYEE,ANSR,1,YES	
2612	013244	104043			EMT	C\$G\$MAN	
2613	013246	000404			BR	10001\$	
2614	013250	002174			.WORD	ANSR	
2615	013252	000130			.WORD	T\$CODE	
2616	013254	014274			.WORD	SYEE	
2617	013256	000001			.WORD	1	
2618	013260						
2619	013260	005737	002174		TST	ANSR	:STANDARD MEMORY RANGE?
2620	013264	001020			BNE	1\$:BR IF YES
2621	013266	005237	002174		INC	ANSR	:RESTORE DEFAULT VALUE
2622	013272				G\$MANID	LOADR,STORE,D,-1,0,30,NO	
2623	013272	104043			EMT	C\$G\$MAN	
2624	013274	000406			BR	10002\$	
2625	013276	002220			.WORD	STORE	
2626	013300	000042			.WORD	T\$CODE	
2627	013302	003004			.WORD	LOADR	
2628	013304	177777			.WORD	-1	

2629	013306	000000				.WORD	T\$LOLIM		
2630	013310	000030				.WORD	T\$HILIM		
2631	013312				10002\$:				
2632	013312	004737	004742			JSR	PC,SETADR	:GET FIRST PAGE ADDRESS	
2633	013316	013737	002170	013552		MOV	LOPAG,LOD4	:STORE LOW PAGE NO.	
2634	013324	000403				BR	LD4	:SKIP NEXT INSTRUCTION	
2635	013326	012737	100600	013552	1\$:	MOV	#100600,LOD4	:STANDARD PAGE = 200,201	16-24K RANGE
2636	013334	013737	013552	177520	LD4:	MOV	LOD4,PCR	:LOAD STARTING ADDRESS	
2637	013342	005037	002176			CLR	RFLAG	:INDICATE EPROM	
2638	013346	012703	002304			MOV	#SYSROM,R3	:POINT TO CHECKWORDS	
2639	013352	013737	013542	002172		MOV	PGCT,COUNTR	:PAGE COUNT	
2640	013360	012337	002226		SYETST:	MOV	(R3)+,CKWD	:GET CHECKWORD FOR THIS PAGE	
2641	013364	004737	004514			JSR	PC,MEMTST	:TEST MEMORY	
2642	013370	005737	002166			TST	REAL	:DOES THIS MEMORY EXIST?	
2643	013374	001450				BEQ	E14	:BR IF NO	
2644	013376	005737	002242			TST	ERRFLG	:ANY ERRORS?	
2645	013402	001421				BEQ	CONT	:BR IF NO	
2646	013404	004737	004356			JSR	PC,VIRTAD	:GET ADDRESS OF ERROR	
2647	013410	005737	002164			TST	BCF	:LOW BYTE PAGE?	
2648	013414	001004				BNE	HBYTE	:BR IF NO	
2649	013416	012737	002677	002240		MOV	#LOBYT,BYTLOC	:SET POINTER FOR ERROR MSG.	
2650	013424	000403				BR	PRIN	:PRINT ERROR MESSAGE	
2651	013426	012737	002741	002240	HBYTE:	MOV	#HIBYT,BYTLOC	:POINTER FOR ERROR MSG.	
2652	013434	022737	000001	002242	PRIN:	CMP	#1,ERRFLG	:CHECKSUM ERROR?	
2653	013442	001411				BEQ	E12	:BR IF YES	
2654	013444	000416				BR	E13	:ELSE CHECKWORD ERROR	
2655	013446	062737	001002	177520	CONT:	ADD	#1002,PCR	:ADJUST PAGE IN PCR	
2656	013454	005337	002172			DEC	COUNTR	:DEC PAGE COUNT	
2657	013460	001337				BNE	SYETST	:LOOP UNTIL FINISHED	
2658	013462					EXIT	TST	:TEST IS FINISHED	
2659	013462	104032				EMT	C\$EXIT		
2660	013464	000642				.WORD	L10065-		
2661	013466				E12:	ERRDF	46,SYSE,CKSME,CKLOOP		
2662	013466	104562				TRAP	T\$ERCODE		
2663	013470	000056				.WORD	46		
2664	013472	014162				.WORD	SYSE		
2665	013474	013554				.WORD	CKSME		
2666	013476					EXIT	TST		
2667	013476	104032				EMT	C\$EXIT		
2668	013500	000626				.WORD	L10065-		
2669	013502				E13:	ERRDF	47,SYSE,CWKDE,CKLOOP		
2670	013502	104562				TRAP	T\$ERCODE		
2671	013504	000057				.WORD	47		
2672	013506	014162				.WORD	SYSE		
2673	013510	013612				.WORD	CWKDE		
2674	013512					EXIT	TST		
2675	013512	104032				EMT	C\$EXIT		
2676	013514	000612				.WORD	L10065-		
2677	013516				E14:	ERRDF	50,SYSE,NONXT,CKLOOP		
2678	013516	104562				TRAP	T\$ERCODE		
2679	013520	000062				.WORD	50		
2680	013522	014162				.WORD	SYSE		
2681	013524	013644				.WORD	NONXT		
2682	013526					EXIT	TST		
2683	013526	104032				EMT	C\$EXIT		
2684	013530	000576				.WORD	L10065-		

2685	013532			ENDSUB
2686	013532			L10072:
2687	013532	104003		EMT C\$ESUB
2688				
2689	013534			EXIT TST
2690	013534	104032		EMT C\$EXIT
2691	013536	000570		.WORD L10065-
2692				
2693	013540	000000		ADDON: .WORD 0
2694	013542	000000		PGCT: .WORD 0
2695	013544	000000		LOD1: .WORD 0
2696	013546	000000		LOD2: .WORD 0
2697	013550	000000		LOD3: .WORD 0
2698	013552	000000		LOD4: .WORD 0
2699				
2700				
2701	013554			BGNMSG CKSME
2702	013554			CKSME::
2703	013554			PRINTB #ERM6,BYTLOC
2704	013554	013746	002240	MOV BYTLOC,-(SP)
2705	013560	012746	013726	MOV #ERM6,-(SP)
2706	013564	012746	000002	MOV #2,-(SP)
2707	013570	010600		MOV SP,R0
2708	013572	104014		EMT C\$PNTB
2709	013574	062706	000006	ADD #6,SP
2710	013600	004737	004356	JSR PC,VIRTAD
2711	013604	004737	004102	JSR PC,VIPRI
2712	013610			ENDMSG
2713	013610			L10073:
2714	013610	104023		EMT C\$MSG
2715				
2716	013612			BGNMSG CWKDE
2717	013612			CWKDE::
2718	013612			PRINTB #ERM5
2719	013612	012746	013676	MOV #ERM5,-(SP)
2720	013616	012746	000001	MOV #1,-(SP)
2721	013622	010600		MOV SP,R0
2722	013624	104014		EMT C\$PNTB
2723	013626	062706	000004	ADD #4,SP
2724	013632	004737	004356	JSR PC,VIRTAD
2725	013636	004737	004102	JSR PC,VIPRI
2726	013642			ENDMSG
2727	013642			L10074:
2728	013642	104023		EMT C\$MSG
2729				
2730	013644			BGNMSG NONXT
2731	013644			NONXT::
2732	013644			PRINTB #LOST
2733	013644	012746	013755	MOV #LOST,-(SP)
2734	013650	012746	000001	MOV #1,-(SP)
2735	013654	010600		MOV SP,R0
2736	013656	104014		EMT C\$PNTB
2737	013660	062706	000004	ADD #4,SP
2738	013664	004737	004356	JSR PC,VIRTAD
2739	013670	004737	004102	JSR PC,VIPRI
2740	013674			ENDMSG

2741	013674				L10075:	
2742	013674	104023			EMT	C\$MSG
2743						
2744	013676	040445	047111	047503	ERM5:	.ASCIZ /%AINCORRECT CHECKWORD%/
2745	013704	051122	041505	020124		
2746	013712	044103	041505	053513		
2747	013720	051117	022504	000116		
2748						
2749	013726	040445	044103	041505	ERM6:	.ASCIZ /%ACHECKSUM ERROR%N%T%/
2750	013734	051513	046525	042440		
2751	013742	051122	051117	047045		
2752	013750	052045	047045	000		
2753						
2754	013755	045	047101	047117	LOST:	.ASCIZ /%ANON-EXISTENT MEMORY%/
2755	013762	042455	044530	052123		
2756	013770	047105	020124	042515		
2757	013776	047515	054522	047045		
2758	014004	000				
2759						
2760	014005	110	053517	046440	RWDCT:	.ASCIZ /HOW MANY CHECKWORDS WILL BE INPUT/
2761	014012	047101	020131	044103		
2762	014020	041505	053513	051117		
2763	014026	051504	053440	046111		
2764	014034	020114	042502	044440		
2765	014042	050116	052125	000		
2766						
2767	014047	101	054516	040440	EXEC:	.ASCIZ /ANY ADDITIONAL MEMORY /
2768	014054	042104	052111	047511		
2769	014062	040516	020114	042515		
2770	014070	047515	054522	000040		
2771						
2772	014076	054105	040520	042116	EXPND:	.ASCIZ /EXPANDED DIAGNOSTIC ROM/
2773	014104	042105	042040	040511		
2774	014112	047107	051517	044524		
2775	014120	020103	047522	000115		
2776						
2777	014126	050105	047522	020115	EPRM:	.ASCIZ /EPROM IN SOCKETS/
2778	014134	047111	051440	041517		
2779	014142	042513	051524	000		
2780						
2781	014147	123	051531	042524	SYSR:	.ASCIZ /SYSTEM ROM/
2782	014154	020115	047522	000115		
2783						
2784	014162	054523	052123	046505	SYSE:	.ASCIZ /SYSTEM EPROM/
2785	014170	042440	051120	046517		
2786	014176	000				
2787						
2788	014177	105	050130	047101	EXADD:	.ASCIZ /EXPANDED ROM IN 2-4K /
2789	014204	042504	020104	047522		
2790	014212	020115	047111	031040		
2791	014220	032055	020113	000		
2792						
2793	014225	105	051120	046517	EPADD:	.ASCIZ /EPROM IN 4-6K /
2794	014232	044440	020116	026464		
2795	014240	045466	000040			
2796						

2797	014244	054523	052123	046505	SRR:	.ASCIZ	/SYSTEM ROM START AT 16K/
2798	014252	051040	046517	051440			
2799	014260	040524	052122	040440			
2800	014266	020124	033061	000113			
2801							
2802	014274	054523	052123	046505	SYEE:	.ASCIZ	/SYSTEM EPROM START AT 16K/
2803	014302	042440	051120	046517			
2804	014310	051440	040524	052122			
2805	014316	040440	020124	033061			
2806	014324	000113					
2807							
2808							
2809							
2810	014326					.EVEN	
2811	014326					ENDTST	
2812	014326	104001			L10065:	EMT	CSETST
2813							
2814							
2815							
2816							
2817							
2818							
2819							
2820							
2821							

PARAMETER CODING MACY11 30A(1052) 28-NOV-78 16:39 PAGE 63 L 5
CVMBAB.P11 28-NOV-78 16:02 TEST 7: TEST ALL ADDITIONAL MEMORY

SEQ 0063

2822
2823
2824
2825

.TITLE PARAMETER CODING
.SBTTL IDENTIFICATION

2826
 2827
 2828
 2829
 2830
 2831
 2832
 2833
 2834
 2835
 2836
 2837
 2838 014330
 2839 014330 000073
 2840 014332
 2841
 2842 014332
 2843 014332 000032
 2844 014334 014404
 2845 014336 160000
 2846 014340 000000
 2847 014342 000016
 2848 014344
 2849 014344 001032
 2850 014346 014420
 2851 014350 177777
 2852 014352 000066
 2853 014354 000100
 2854 014356
 2855 014356 002032
 2856 014360 014451
 2857 014362 177777
 2858 014364 000000
 2859 014366 000007
 2860 014370
 2861 014370 003032
 2862 014372 014471
 2863 014374 177777
 2864 014376 000000
 2865 014400 007777
 2866
 2867 014402
 2868 014402 047004
 2869
 2870 014404 047125 052111 047040
 2871 014412 046525 042502 000122
 2872 014420 047111 042524 051122
 2873 014426 050125 020124 042526
 2874 014434 052103 051117 040440
 2875 014442 042104 042522 051523
 2876 014450 000
 2877 014451 111 052116 051105
 2878 014456 052522 052120 046040
 2879 014464 053105 046105 000
 2880 014471 122 041517 042513
 2881 014476 020122 053523 052111

.SBTTL HARDWARE PARAMETER CODING SECTION

```

:++
: 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.
:--

```

```

BGNHRD
.WORD L10076-L$HARD/2
L$HARD::

GPRMD UNIT,0,0,160000,0,16,YES
.WORD T$CODE
.WORD UNIT
.WORD 160000
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD INTVEC,2,0,-1,66,100,YES
.WORD T$CODE
.WORD INTVEC
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD PRI,4,0,-1,0,7,YES
.WORD T$CODE
.WORD PRI
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD RKSW,6,0,-1,0,7777,YES
.WORD T$CODE
.WORD RKSW
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM

```

```

EXIT HRD
.WORD T$CODE

UNIT: .ASCIZ /UNIT NUMBER/

INTVEC: .ASCIZ /INTERRUPT VECTOR ADDRESS/

PRI: .ASCIZ /INTERRUPT LEVEL/

RKSW: .ASCIZ /ROCKER SWITCH SETTINGS/

```

2882	014504	044103	051440	052105	
2883	014512	044524	043516	000123	
2884					.EVEN
2885					
2886	014520				ENDHRD
2887					.EVEN
2888	014520				L10076:

2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900 014520
2901 014520 000161
2902 014522
2903 014522
2904 014522 000032
2905 014524 014644
2906 014526 177777
2907 014530 000000
2908 014532 177777
2909 014534
2910 014534 001032
2911 014536 014721
2912 014540 177777
2913 014542 000000
2914 014544 177777
2915 014546
2916 014546 002032
2917 014550 014737
2918 014552 177777
2919 014554 000000
2920 014556 177777
2921 014560
2922 014560 003032
2923 014562 014755
2924 014564 177777
2925 014566 000000
2926 014570 177777
2927 014572
2928 014572 004032
2929 014574 014773
2930 014576 177777
2931 014600 000000
2932 014602 177777
2933 014604
2934 014604 005032
2935 014606 015011
2936 014610 177777
2937 014612 000000
2938 014614 177777
2939 014616
2940 014616 006032
2941 014620 015027
2942 014622 177777
2943 014624 000000
2944 014626 177777

.SBTTL SOFTWARE PARAMETER CODING SECTION

```

:++
: 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.
:--

```

```

BGNSFT
.WORD L10077-L$SOFT/2
L$SOFT::
GPRMD CKW1,0,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW1
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD CKW2,2,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW2
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD CKW3,4,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW3
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD CKW4,6,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW4
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD CKW5,10,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW5
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD CKW6,12,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW6
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM
GPRMD CKW7,14,0,-1,0,177777,YES
.WORD T$CODE
.WORD CKW7
.WORD -1
.WORD T$LLOLIM
.WORD T$HILIM

```

```

2945 014630          GPRMD  CKW8,16,0,-1,0,177777,YES
2946 014630 007032  .WORD  T$CODE
2947 014632 015045  .WORD  CKW8
2948 014634 177777  .WORD  -1
2949 014636 000000  .WORD  T$LLOLIM
2950 014640 177777  .WORD  T$HILIM
2951
2952 014642          EXIT SFT
2953 014642 111004  .WORD  T$CODE
2954
2955 014644 044103 041505 053513 CKW1:  .ASCIZ  /CHECKWORDS FOR DIAGNOSTIC ROM. CHECKWORD 1: /
2956 014652 051117 051504 043040
2957 014660 051117 042040 040511
2958 014666 047107 051517 044524
2959 014674 020103 047522 027115
2960 014702 041440 042510 045503
2961 014710 047527 042122 030440
2962 014716 020072 000
2963 014721 103 042510 045503 CKW2:  .ASCIZ  /CHECKWORD 2: /
2964 014726 047527 042122 031040
2965 014734 020072 000
2966 014737 103 042510 045503 CKW3:  .ASCIZ  /CHECKWORD 3: /
2967 014744 047527 042122 031440
2968 014752 020072 000
2969 014755 103 042510 045503 CKW4:  .ASCIZ  /CHECKWORD 4: /
2970 014762 047527 042122 032040
2971 014770 020072 000
2972 014773 103 042510 045503 CKW5:  .ASCIZ  /CHECKWORD 5: /
2973 015000 047527 042122 032440
2974 015006 020072 000
2975 015011 103 042510 045503 CKW6:  .ASCIZ  /CHECKWORD 6: /
2976 015016 047527 042122 033040
2977 015024 020072 000
2978 015027 103 042510 045503 CKW7:  .ASCIZ  /CHECKWORD 7: /
2979 015034 047527 042122 033440
2980 015042 020072 000
2981 015045 103 042510 045503 CKW8:  .ASCIZ  /CHECKWORD 8: /
2982 015052 047527 042122 034040
2983 015060 020072 000
2984 015064          .EVEN
2985
2986
2987 015064          ENDSFT
2988          .EVEN
2989 015064          L10077:
2990
2991          015134          .=.+50
2992 015134          LASTAD
2993          .EVEN
2994 015134          L$LAST::
    
```



```
2995  
2996 045730 000000 .SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP  
045732 000000 .WORD 0 ;SPACE FOR USER POOL POINTER  
045734 000000 .WORD 0 ;SIZE  
045736 000000 .WORD 0 ;CHECKSUM (NOT CURRENTLY USED)  
045742 .END.SUPV=+.2 ;SIZE OF H.W. PTAB. ALLOCATION  
000200 .END 200
```


CKW3	014737	2917	2966#																	
CKW4	014755	2923	2969#																	
CKW5	014773	2929	2972#																	
CKW6	015011	2935	2975#																	
CKW7	015027	2941	2978#																	
CKW8	015045	2947	2981#																	
CLEAR.	023522	2996#																		
CLKACC	015374	G	2996#*																	
CLKBFR	042624	G	2996#*																	
CLKCNT	015372	G	2996#*																	
CLKJUM	043230	G	2996#																	
CLKRES	044232	G	2996#																	
CLKSER	044366	G	2996#																	
CLKSON	015432	G	2996#*																	
CLK.SE	022202		2996#																	
CLR.MA	022456		2996#																	
CNVT	040700		2996#																	
COMMAN	015204	G	2996#*																	
COMMTA	040514		2996#																	
CONT	013446		2645	2655#																
CONTCL	044312	G	2996#																	
COUNTR	002172		558#	2026*	2104*	2378*	2395*	2460*	2477*	2550*	2567*	2639*	2656*							
CRLF	034612		2996#																	
CURR.S	015142	G	2996#*																	
CURR.T	015144	G	2996#*																	
CWDERR	002653		627#																	
CKWDE	013612	G	2414	2493	2582	2673	2717#													
CSAAD	027502		2996#																	
CSAAE	027514		2996#																	
CSAAK	030512		2996#																	
CSAAL	030656		2996#																	
CSABRT=	000021		359#																	
CSADR =	000020		359#	1206	1227	1248	1269	1288	1309	1330	1351	1376	1403	1434	1455					
			1477	1498	1519	1539	1559	1580	1605	1632	1666	1698	1716	1871	2040					
			2496	2573	2579	2585	2662	2670	2678											
CSAU =	000054		359#																	
CSBRK =	000022		359#																	
CSBSEG=	000004		359#																	
CSBSUB=	000002		359#	1200	1220	1241	1262	1283	1302	1323	1344	1365	1390	1428	1448					
			1470	1491	1512	1532	1552	1573	1594	1619	1666	1698	1994	2058	2087					
			2102	2264	2345	2427	2506	2595												
CSBUFF =	000030		359#																	
CSCEFG=	000046		359#																	
CSCLEA=	000012		359#	1183																
CSCLP1=	000006		359#	981	1214	1235	1256	1277	1296	1317	1338	1359	1384	1411	1442					
			1463	1485	1506	1527	1547	1567	1588	1613	1640	1691	1716	1871	2040					
			2051	2070	2081	2095														
CSVEC=	000036		359#	1173																
CSDECLN=	000044		359#																	
CSDODU=	000053		359#																	
CSDRPT=	000024		359#																	
CSDU =	000055		359#																	
CSEDIT=	000002		359#	425																
CSERDF=	000002		359#	977	1206	1227	1248	1269	1288	1309	1330	1351	1376	1403	1434					
			1455	1477	1498	1519	1539	1559	1580	1605	1632	1687	1712	1867	2033					
			2047	2063	2077	2091	2405	2411	2417	2484	2490	2496	2573	2579	2585					

EXPND	014076	2298	2407	2413	2419	2772#								
EXPROM	011770	2302	2346#											
EXPSUM	002200	561#	1007*	1008*	1009	1024*	1025*	1026	2042*	2043*	2044	2072*	2073*	2074
EXPTST	012130	2379#	2396											
E1	012254	2392	2404#											
E10	013144	2565	2578#											
E11	013156	2554	2584#											
E12	013466	2653	2661#											
E13	013502	2654	2669#											
E14	013516	2643	2677#											
E2	012266	2393	2410#											
E3	012300	2382	2416#											
E4	012574	2474	2483#											
E5	012606	2475	2489#											
E6	012620	2464	2495#											
E7	013132	2564	2572#											
FILL	035260	2996#												
FILL.C	000204	2996#												
FLAGS	015200	2996#*												
FLAGS1	015202	2996#												
FLAGTA	040432	2996#												
FLAG.I	022166	2996#												
FLA.SE	040400	2996#												
FLG.MA	022126	2996#												
FORM.T	031030	2996#*												
FREE	036336	2996#												
F\$AU =	000015	359#												
F\$BGN =	000040	359#	369	456	465	477	540	546	554	583	749	769	789	809
		829	841	853	1100	1114	1137	1161	1176	1198	1200	1211	1216	1220
		1232	1237	1241	1253	1258	1262	1274	1279	1283	1293	1298	1302	1314
		1319	1323	1335	1340	1344	1356	1361	1365	1381	1386	1390	1408	1413
		1417	1426	1428	1439	1444	1448	1460	1465	1470	1482	1487	1491	1503
		1508	1512	1524	1529	1532	1544	1549	1552	1564	1569	1573	1585	1590
		1594	1610	1615	1619	1637	1642	1645	1657	1663	1666	1694	1698	1719
		1722	1726	1735	1754	1797	1806	1822	1842	1846	1859	1873	1935	1943
		1981	1992	1994	2037	2053	2058	2067	2083	2087	2097	2102	2107	2111
		2117	2130	2143	2156	2169	2235	2262	2264	2337	2340	2345	2402	2423
		2427	2502	2506	2570	2591	2595	2659	2667	2675	2683	2686	2690	2702
		2717	2731	2811	2839	2868	2901	2953						
		359#	1161	1182										
		359#												
		359#	369	456	465	477	540	546	554	583	767	787	807	827
		839	851	863	1104	1137	1153	1176	1184	1198	1200	1211	1216	1218
		1220	1232	1237	1239	1241	1253	1258	1260	1262	1274	1279	1281	1283
		1293	1298	1300	1302	1314	1319	1321	1323	1335	1340	1342	1344	1356
		1361	1363	1365	1381	1386	1388	1390	1408	1413	1415	1417	1419	1426
		1428	1439	1444	1446	1448	1460	1465	1467	1470	1482	1487	1489	1491
		1503	1508	1510	1512	1524	1529	1531	1532	1544	1549	1551	1552	1564
		1569	1571	1573	1585	1590	1592	1594	1610	1615	1617	1619	1637	1642
		1644	1645	1647	1657	1663	1666	1694	1696	1698	1719	1721	1722	1731
		1752	1771	1797	1799	1806	1822	1842	1846	1848	1859	1873	1935	1955
		1981	1983	1992	1994	2037	2053	2055	2058	2067	2083	2085	2087	2097
		2099	2102	2107	2109	2111	2128	2141	2154	2167	2180	2235	2237	2262
		2264	2337	2340	2342	2345	2402	2423	2425	2427	2502	2504	2506	2570
		2591	2593	2595	2659	2667	2675	2683	2686	2688	2690	2715	2729	2743
		2811	2813	2868	2889	2953	2990							

F\$CLEA= 000007
 F\$DU = 000016
 F\$END = 000041

L\$AUT	002074	G	443#		
L\$CCP	002106	G	453#		
L\$CLEA	005200	G	454	1161#	
L\$CO	002032	G	409#		
L\$DEPO	002011	G	391#		
L\$DESC	002102	G	449#		
L\$DEVP	002064	G	435#		
L\$DISP	002112	G	416	468#	
L\$DR	002506	G	440	589#	
L\$DRCT	002070	G	439#		
L\$DRS	002072	G	441#		
L\$DRST	002512	G	442	593#	
L\$DTP	002040	G	415#		
L\$DUT	002076	G	445#		
L\$DVTY	003036	G	436	659#	
L\$EF	002056	G	430#		
L\$EFLG	002034	G	411#		
L\$EXP1	002042	G	417#		
L\$EXP2	002044	G	419#		
L\$EXP3	002046	G	421#		
L\$HARD	014332	G	398	2839	2840#
L\$HPCP	002016	G	397#		
L\$HPTP	002022	G	401#		
L\$HW	002132	G	402	487	488#
L\$ICP	002104	G	451#		
L\$INIT	005034	G	452	1114#	
L\$LDAP	002026	G	405#		
L\$LAST	015134	G	406	2994#	
L\$MREV	002050	G	423#		
L\$NAME	002000	G	380#		
L\$REPP	002066	G	437#		
L\$REV	002010	G	389#		
L\$RPT	005032	G	1100#		
L\$SOFT	014522	G	400	2901	2902#
L\$SPC	002062	G	433#		
L\$SPCP	002020	G	399#		
L\$SPTP	002024	G	403#		
L\$STA	002030	G	407#		
L\$SW	002144	G	404	508	509#
L\$TIML	002014	G	395#		
L\$TIMU	002054	G	428#		
L\$TIM1	002052	G	426#		
L\$TSTI	002100	G	447#		
L\$UNIT	002012	G	393#		
L.CLK.	021532		2996#		
L1000	002142		487	498#	
L1001	002164		508	525#	
L1002	003630		765#		
L1003	003676		785#		
L1004	003744		805#		
L1005	004012		825#		
L1006	004034		837#		
L1007	004056		849#		
L10010	004100		861#		
L10011	005032		1102#		
L10012	005176		1138	1151#	

RSET	002232	574#	1165*	2004	2009														
RSTACK	044560	G	2996#																
RSTRT	010476		2000	2012	2025#														
RWDCT	014005		2513	2602	2760#														
RWR	002522		608#	1208	1229	1250	1271	1290	1311	1332	1353	1378	1405						
RWREG =	177522		1164*	1195#	1201*	1204	1221*	1222	1225	1242*	1243	1246	1263*	1264	1267				
			1284*	1285	1303*	1304	1307	1324*	1325	1328	1345*	1346	1349	1366*	1367*				
			1368	1369	1373*	1391*	1392*	1393	1394	1397*									
SAVEDO=	017070		2996#																
SEARCH	036566		2996#																
SEGSTA	015434	G	2996#*																
SERR1	010260		1947	1966#															
SETADR	004742		1054#	2023	2371	2453	2543	2632											
SET.MA	022312		2996#																
SFPTBL	002144	G	510#	2027															
SHIFT	045416	G	2996#																
SPEC.U	022026		2996#*																
SPV.SE	000400		2996#																
SRR	014244		2527	2797#															
STARTC	044306	G	2996#																
STORE	002220		569#	926	933	1054	2016	2364	2446	2536	2625								
STRCHR	035320		2996#																
STRT	010434		1998	2011#															
STRT.T	022104		2996#*																
ST.SET	016736		2996#																
SUNIT.	022110		2996#*																
SUPERV	017772		2996#																
SUPFLA	015400	G	2996#*																
SUPV.T	015552	G	2996#*																
SUP.PR	016510		2996#																
SVC.GBL =	000000		359#	360#	369	370	380	381	389	390	391	392	393	394	395				
			396	397	398	399	400	401	402	403	404	405	406	407	408				
			409	410	411	412	413	414	415	416	417	418	419	420	421				
			422	423	424	426	427	428	429	430	431	433	434	435	436				
			437	438	439	440	441	442	443	444	445	446	447	448	449				
			450	451	452	453	454	465	466	468	469	488	489	490	509				
			510	511	540	541	554	555	589	590	591	593	594	595	659				
			660	749	750	769	770	789	790	809	810	829	830	841	842				
			853	854	1100	1101	1114	1115	1161	1162	1726	1727	1735	1736	1754				
			1755	1943	1944	2117	2118	2130	2131	2143	2144	2156	2157	2169	2170				
			2702	2703	2717	2718	2731	2732	2840	2841	2902	2903	2994#	2995					
SVCHAN	024426		2996#																
SVCINS =	000000		359#	381	382	383	384	385	386	387	388	389	390	391	392				
			393	394	395	396	397	398	399	400	401	402	403	404	405				
			406	407	408	409	410	411	412	413	414	415	416	417	418				
			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	455	467	468				
			469	470	471	472	473	474	475	476	487	488	508	509	588				
			589	591	592	593	595	596	660	662	663	751	752	753	754				
			755	756	757	758	759	760	761	762	763	764	766	767	771				
			772	773	774	775	776	777	778	779	780	781	782	783	784				
			786	787	791	792	793	794	795	796	797	798	799	800	801				
			802	803	804	806	807	811	812	813	814	815	816	817	818				
			819	820	821	822	823	824	826	827	831	832	833	834	835				
			836	838	839	843	844	845	846	847	848	850	851	855	856				

857	858	859	860	862	863	867	868	869	870	871	872	873
874	918	919	920	921	922	923	924	925	926	927	928	929
930	931	932	977	978	979	980	981	982	1103	1104	1116	1117
1118	1119	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133
1134	1135	1137	1138	1139	1152	1153	1172	1173	1174	1176	1177	1178
1183	1184	1200	1201	1206	1207	1208	1209	1210	1211	1212	1213	1214
1215	1217	1218	1220	1221	1227	1228	1229	1230	1231	1232	1233	1234
1235	1236	1238	1239	1241	1242	1248	1249	1250	1251	1252	1253	1254
1255	1256	1257	1259	1260	1262	1263	1269	1270	1271	1272	1273	1274
1275	1276	1277	1278	1280	1281	1283	1284	1288	1289	1290	1291	1292
1293	1294	1295	1296	1297	1299	1300	1302	1303	1309	1310	1311	1312
1313	1314	1315	1316	1317	1318	1320	1321	1323	1324	1330	1331	1332
1333	1334	1335	1336	1337	1338	1339	1341	1342	1344	1345	1351	1352
1353	1354	1355	1356	1357	1358	1359	1360	1362	1363	1365	1366	1376
1377	1378	1379	1380	1381	1382	1383	1384	1385	1387	1388	1390	1391
1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1414	1415	1418
1419	1428	1429	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443
1445	1446	1448	1449	1455	1456	1457	1458	1459	1460	1461	1462	1463
1464	1466	1467	1470	1471	1477	1478	1479	1480	1481	1482	1483	1484
1485	1486	1488	1489	1491	1492	1498	1499	1500	1501	1502	1503	1504
1505	1506	1507	1509	1510	1512	1513	1519	1520	1521	1522	1523	1524
1525	1526	1527	1528	1530	1531	1532	1533	1539	1540	1541	1542	1543
1544	1545	1546	1547	1548	1550	1551	1552	1553	1559	1560	1561	1562
1563	1564	1565	1566	1567	1568	1570	1571	1573	1574	1580	1581	1582
1583	1584	1585	1586	1587	1588	1589	1591	1592	1594	1595	1605	1606
1607	1608	1609	1610	1611	1612	1613	1614	1616	1617	1619	1620	1632
1633	1634	1635	1636	1637	1638	1639	1640	1641	1643	1644	1646	1647
1663	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1676
1677	1678	1679	1680	1681	1682	1683	1684	1687	1688	1689	1690	1691
1692	1695	1696	1698	1699	1701	1702	1703	1704	1705	1706	1707	1708
1709	1712	1713	1714	1715	1716	1717	1720	1721	1722	1723	1724	1730
1731	1737	1738	1739	1740	1741	1742	1743	1744	1745	1746	1747	1748
1749	1751	1752	1756	1757	1758	1759	1760	1761	1762	1763	1764	1765
1766	1767	1768	1770	1771	1798	1799	1809	1810	1811	1813	1814	1815
1816	1817	1818	1819	1822	1823	1824	1827	1828	1829	1831	1832	1833
1835	1836	1837	1839	1840	1841	1842	1843	1844	1847	1848	1861	1862
1863	1864	1867	1868	1869	1870	1871	1872	1873	1874	1875	1891	1892
1893	1894	1895	1896	1897	1903	1904	1905	1906	1907	1908	1909	1917
1918	1919	1920	1921	1922	1923	1928	1929	1930	1931	1932	1933	1935
1936	1937	1945	1946	1947	1948	1949	1950	1951	1952	1954	1955	1982
1983	1994	1995	1996	1997	1998	1999	2002	2003	2004	2005	2006	2007
2008	2014	2015	2016	2017	2018	2019	2020	2021	2022	2033	2034	2035
2036	2037	2038	2039	2040	2041	2047	2048	2049	2050	2051	2052	2054
2055	2058	2059	2063	2064	2065	2066	2067	2068	2069	2070	2071	2077
2078	2079	2080	2081	2082	2084	2085	2087	2088	2091	2092	2093	2094
2095	2096	2098	2099	2102	2103	2108	2109	2111	2112	2113	2119	2120
2121	2122	2123	2124	2125	2127	2128	2132	2133	2134	2135	2136	2137
2138	2140	2141	2145	2146	2147	2148	2149	2150	2151	2153	2154	2158
2159	2160	2161	2162	2163	2164	2166	2167	2171	2172	2173	2174	2175
2176	2179	2180	2236	2237	2264	2265	2266	2267	2268	2269	2284	2285
2286	2287	2288	2289	2290	2294	2295	2296	2297	2298	2299	2300	2304
2305	2306	2307	2308	2309	2310	2315	2316	2317	2318	2319	2320	2321
2326	2327	2328	2329	2330	2331	2332	2337	2338	2339	2341	2342	2345
2346	2351	2352	2353	2354	2355	2356	2357	2362	2363	2364	2365	2366
2367	2368	2369	2370	2402	2403	2404	2405	2406	2407	2408	2409	2411
2412	2413	2414	2415	2417	2418	2419	2420	2421	2424	2425	2427	2428

SSLSYM= 010000	359#	499#	526#	766#	786#	806#	826#	838#	850#	862#	925	932	933#
	1103#	1152#	1183#	1217#	1238#	1259#	1280#	1299#	1320#	1341#	1362#	1387#	1414#
	1418#	1445#	1466#	1488#	1509#	1530#	1550#	1570#	1591#	1616#	1643#	1646#	1695#
	1720#	1730#	1751#	1770#	1798#	1847#	1954#	1982#	2003	2008	2009#	2015	2022
	2023#	2054#	2084#	2098#	2108#	2127#	2140#	2153#	2166#	2179#	2236#	2285	2290
	2291#	2295	2300	2301#	2305	2310	2311#	2316	2321	2322#	2327	2332	2333#
	2341#	2352	2357	2358#	2363	2370	2371#	2424#	2434	2439	2440#	2445	2452
	2453#	2503#	2510	2517	2518#	2524	2529	2530#	2535	2542	2543#	2592#	2599
	2606	2607#	2613	2618	2619#	2624	2631	2632#	2687#	2714#	2728#	2742#	2812#
	2889#	2990#											
TAG1 010002	1900#	1912											
TAG2 010032	1901	1909#											
TAG3 010050	1914#	1926											
TAG4 010100	1915	1923#											
TEMP 010140	1875*	1878	1883*	1939#									
TERMI 042616	2996#												
TERMLI 040420	2996#												
TERMTA 034402	2996#												
TEST.M 022040	2996#*												
TIMFLG 015370	2996#*												
TIM.CO 015222	2996#*												
TIM.OP 031026	2996#*												
TOO.MA 034362	2996#												
TSTCKW 004672	1017	1027	1031#										
TST.AB 024350	2996#												
TST.TO 016552	2996#												
TYPEC 034756	2996#												
TYPEPC 030652	2996#												
TYPFLA 040274	2996#												
TYPLIN 034654	2996#												
TYPNUM 034236	2996#												
TYPSTR 034674	2996#												
TYP.ER 030502	2996#												
TY.UNI 023514	2996#												
T\$ARGC= 000001	381#	382#	383#	384#	385#	386#	387#	751#	755	757#	763	771#	775
	777#	783	791#	795	797#	803	811#	815	817#	823	831#	835	843#
	847	855#	859	867#	873	918#	922	1130#	1134	1737#	1741	1743#	1748
	1756#	1760	1762#	1767	1891#	1896	1903#	1908	1917#	1922	1928#	1932	1945#
	1951	2119#	2124	2132#	2137	2145#	2150	2158#	2163	2171#	2175	2704#	2709
	2719#	2723	2733#	2737									
T\$CODE= 111004	927#	2005#	2017#	2287#	2297#	2307#	2318#	2329#	2354#	2365#	2436#	2447#	2512#
	2526#	2537#	2601#	2615#	2626#	2843#	2849#	2855#	2861#	2868#	2904#	2910#	2916#
T\$ERCO= 000162	2922#	2928#	2934#	2940#	2946#	2953#							
	977#	1206#	1227#	1248#	1269#	1288#	1309#	1330#	1351#	1376#	1403#	1434#	1455#
	1477#	1498#	1519#	1539#	1559#	1580#	1605#	1632#	1687#	1712#	1867#	2033#	2047#
	2063#	2077#	2091#	2405#	2411#	2417#	2484#	2490#	2496#	2573#	2579#	2585#	2662#
	2670#	2678#											
T\$ERRN= 000062	359#	978#	1207#	1228#	1249#	1270#	1289#	1310#	1331#	1352#	1377#	1404#	1435#
	1456#	1478#	1499#	1520#	1540#	1560#	1581#	1606#	1633#	1688#	1713#	1868#	2034#
	2048#	2064#	2078#	2092#	2406#	2412#	2418#	2485#	2491#	2497#	2574#	2580#	2586#
	2663#	2671#	2679#										
T\$EXCP= 000000	927#	932	2017#	2022	2365#	2370	2447#	2452	2512#	2517	2537#	2542	2601#
	2606	2626#	2631	2843#	2848	2849#	2854	2855#	2860	2861#	2866	2904#	2909
	2910#	2915	2916#	2921	2922#	2927	2928#	2933	2934#	2939	2940#	2945	2946#
	2951												
T\$FLAG= 000041	1137#	1176#	1211#	1232#	1253#	1274#	1293#	1314#	1335#	1356#	1381#	1408#	1439#

	1460#	1482#	1503#	1524#	1544#	1564#	1585#	1610#	1637#	1663#	1722#	1822#	1842#
	1873#	1935#	2037#	2067#	2111#	2337#	2402#	2570#	2659#	2667#	2675#	2683#	2690#
T\$HILI= 177777	2868#	2953#											
	927#	931	2017#	2021	2365#	2369	2447#	2451	2512#	2516	2537#	2541	2601#
	2605	2626#	2630	2843#	2847	2849#	2853	2855#	2859	2861#	2865	2904#	2908
	2910#	2914	2916#	2920	2922#	2926	2928#	2932	2934#	2938	2940#	2944	2946#
	2950												
T\$LOLI= 000000	927#	930	2017#	2020	2365#	2368	2447#	2450	2512#	2515	2537#	2540	2601#
	2604	2626#	2629	2843#	2846	2849#	2852	2855#	2858	2861#	2864	2904#	2907
	2910#	2913	2916#	2919	2922#	2925	2928#	2931	2934#	2937	2940#	2943	2946#
	2949												
T\$LSYM= 010000	359#	499	526	766	786	806	826	838	850	862	1103	1152	1183
	1217	1238	1259	1280	1299	1320	1341	1362	1387	1414	1418	1445	1466
	1488	1509	1530	1550	1570	1591	1616	1643	1646	1695	1720	1730	1751
	1770	1798	1847	1954	1982	2054	2084	2098	2108	2127	2140	2153	2166
T\$NEST= 177777	2179	2236	2341	2424	2503	2592	2687	2714	2728	2742	2812	2889	2990
	359#	370#	456#	466#	477#	487#	498#	508#	525#	541#	546#	555#	583#
	749#	765#	769#	785#	789#	805#	809#	825#	829#	837#	841#	849#	853#
	861#	1100#	1102#	1114#	1151#	1161#	1182#	1198#	1200#	1216#	1220#	1237#	1241#
	1258#	1262#	1279#	1283#	1298#	1302#	1319#	1323#	1340#	1344#	1361#	1365#	1386#
	1390#	1413#	1417#	1426#	1428#	1444#	1448#	1465#	1470#	1487#	1491#	1508#	1512#
	1529#	1532#	1549#	1552#	1569#	1573#	1590#	1594#	1615#	1619#	1642#	1645#	1657#
	1666#	1694#	1698#	1719#	1726#	1729#	1735#	1750#	1754#	1769#	1797#	1806#	1846#
	1859#	1943#	1953#	1981#	1992#	1994#	2053#	2058#	2083#	2087#	2097#	2102#	2107#
	2117#	2126#	2130#	2139#	2143#	2152#	2156#	2165#	2169#	2178#	2235#	2262#	2264#
	2340#	2345#	2423#	2427#	2502#	2506#	2591#	2595#	2686#	2702#	2713#	2717#	2727#
T\$NSK0= 000005	2731#	2741#	2811#	2839#	2868	2887#	2901#	2953	2988#				
	370#	456	466#	477	487#	498	508#	525	541#	546	555#	583	749#
	765	769#	785	789#	805	809#	825	829#	837	841#	849	853#	861
	1100#	1102	1114#	1151	1161#	1182	1198#	1417	1426#	1645	1657#	1797	1806#
T\$NSK1= 000011	1846	1859#	1981	1992#	2235	2262#	2811	2839#	2868	2887	2901#	2953	2988
	1200#	1216	1220#	1237	1241#	1258	1262#	1279	1283#	1298	1302#	1319	1323#
	1340	1344#	1361	1365#	1386	1390#	1413	1428#	1444	1448#	1465	1470#	1487
	1491#	1508	1512#	1529	1532#	1549	1552#	1569	1573#	1590	1594#	1615	1619#
	1642	1666#	1694	1698#	1719	1726#	1729	1735#	1750	1754#	1769	1943#	1953
	1994#	2053	2058#	2083	2087#	2097	2102#	2107	2117#	2126	2130#	2139	2143#
	2152	2156#	2165	2169#	2178	2264#	2340	2345#	2423	2427#	2502	2506#	2591
	2595#	2686	2702#	2713	2717#	2727	2731#	2741					
T\$SAVL= 177777	359#												
T\$SEGL= 177777	359#												
T\$SUBN= 000005	359#	1198#	1200#	1220#	1241#	1262#	1283#	1302#	1323#	1344#	1365#	1390#	1426#
	1428#	1448#	1470#	1491#	1512#	1532#	1552#	1573#	1594#	1619#	1657#	1666#	1698#
	1806#	1859#	1992#	1994#	2058#	2087#	2102#	2262#	2264#	2345#	2427#	2506#	2595#
T\$TAGL= 177777	359#												
T\$TAGN= 010100	359#	487#	508#	749#	769#	789#	809#	829#	841#	853#	1100#	1114#	1161#
	1198#	1200#	1220#	1241#	1262#	1283#	1302#	1323#	1344#	1365#	1390#	1426#	1428#
	1448#	1470#	1491#	1512#	1532#	1552#	1573#	1594#	1619#	1657#	1666#	1698#	1726#
	1735#	1754#	1806#	1859#	1943#	1992#	1994#	2058#	2087#	2102#	2117#	2130#	2143#
	2156#	2169#	2262#	2264#	2345#	2427#	2506#	2595#	2702#	2717#	2731#	2839#	2901#
T\$TEMP= 000005	456#	469#	470#	471#	472#	473#	474#	475#	476#	477#	498#	525#	546#
	583#	765#	785#	805#	825#	837#	849#	861#	927#	1102#	1137#	1138	1151#
	1176#	1177	1182#	1211#	1212	1216#	1232#	1233	1237#	1253#	1254	1258#	1274#
	1275	1279#	1293#	1294	1298#	1314#	1315	1319#	1335#	1336	1340#	1356#	1357
	1361#	1381#	1382	1386#	1408#	1409	1413#	1417#	1439#	1440	1444#	1460#	1461
	1465#	1482#	1483	1487#	1503#	1504	1508#	1524#	1525	1529#	1544#	1545	1549#
	1564#	1565	1569#	1585#	1586	1590#	1610#	1611	1615#	1637#	1638	1642#	1645#

XEQ.OP	023730	2996#																
XEQ.PR	017130	2996#																
XEQ.TE	023774	2996#																
XTIME	043316	2996#																
XTIMEN	044142	2996#																
XTIMST	043340	2996#*																
XXDP.D	021504	2996#																
XSALWA=	000000	359#	2868		2953													
XSALS=	000040	359#																
XSOFFS=	000400	359#	2868		2953													
XSTRUE=	000020	359#																
ZERR	003046	670#	751															
\$BREG	022200	2996#*																
\$ENDAD	044416	2996#																
\$SAV2	045462	2996#																
\$SAV3	045476	2996#																
\$SAV4	045514	2996#																
\$SAV5	045534	2996#																
.	= 045740	2#	579#	580#	581#	595#	1138	1177	1212	1233	1254	1275	1294	1315				
		1336	1357	1382	1409	1440	1461	1483	1504	1525	1545	1565	1586	1611				
		1638	1664	1723	1795#	1823	1843	1874	1936	1979#	2038	2068	2112	2338				
		2403	2571	2660	2668	2676	2684	2691	2868	2953	2984#	2991#	2996#					

. ABS. 045740 000

ERRORS DETECTED: 0

DSKZ:CVMBAB,DSKZ:CVMBAB/SOL/CRF:SYM/NL:TOC=CVMBAB/ML, CVMBAB.P11, CVMBAB.SUP
RUN-TIME: 41 38 3 SECONDS
RUN-TIME RATIO: 145/83=1.7
CORE USED: 15K (29 PAGES)